Proficiency in Basic Data Structures among Various Subpopulations of Students at Different Stages in a CS Program

Sander Valstar, Sophia Krause-Levy, Adrian Salguero, Leo Porter and William G. Griswold
University of California, San Diego
{avalstar,skrausel,asalguer,leporter,wgg}@ucsd.edu

ABSTRACT

Previous studies show that CS students may not learn as much from their courses as we might expect. This could have ramifications on how students succeed in their future careers and may explain why researchers report a gap between industry expectations and the abilities of recent CS graduates. However, previous studies have also shown that students improve their prerequisite knowledge in subsequent courses. This study investigates the introductory data structures proficiency of students in different courses at various stages in our CS program, employing the validated Basic Data Structures Inventory (BDSI). Additionally, we investigate whether subpopulations, including transfer students and underrepresented groups, may be more prone to not attaining as much knowledge from our courses as we might expect.

We find that students' knowledge of basic data structures is, on average, better in later courses. However, we also find subpopulations of students that perform worse than others or seem to not improve their knowledge in later courses. Specifically, we find students that transferred to our institution from a different school perform significantly worse on the BDSI than other students and do not improve their BDSI performance in later courses. We also find students from demographic backgrounds that are underrepresented in computing scored slightly, though not statistically significantly, worse than others. Our findings warrant future investigations on how our programs can better serve the students in the affected subpopulations.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; **Computer science education**; **Software engineering education**.

KEYWORDS

CS education, prerequisites, Data Structures, BDSI

ACM Reference Format:
Sander Valstar, Sophia Krause-Levy, Adrian Salguero, Leo Porter and William G. Griswold. 2021. Proficiency in Basic Data Structures among Various Subpopulations of Students at Different Stages in a CS Program. In *26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2021), June 26-July 1, 2021, Virtual Event, Germany.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3430665.3456337

This work is licensed under a Creative Commons Attribution International 4.0 License.

ITiCSE 2021, June 26-July 1, 2021, Virtual Event, Germany.
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8214-4/21/06.
https://doi.org/10.1145/3430665.3456337

1 INTRODUCTION

Inspired by the recent findings that a surprisingly large percentage of students enter later CS classes unable to demonstrate knowledge from earlier classes (prerequisite knowledge) [13, 27], we sought to gain better insight into how students' understanding of a core computing topic changes in a progression of courses in our curriculum. To this end, we administered the validated Basic Data Structures Inventory (BDSI) [19] in three classes in a sequence of dependent courses at our institution to see 1) how much student understanding varies and 2) whether their understanding improves in later courses.

Similar to our study on prerequisite knowledge [27], we found that student performance varied widely on the BDSI, which might pose challenges for instructors trying to address the pedagogical needs of the students in those classes. Encouragingly, we found that student performance on the BDSI increased for each course in the sequence. This increase suggests that either students are improving in their knowledge of basic data structures over time or students who struggled are leaving. If students are improving on their knowledge from prior courses, this is an interesting finding as a previous study documented notable knowledge loss of CS1 material over the period of a summer [24]. However, unlike that study, we believe basic data structure concepts are likely reinforced in each subsequent course rather than being left to atrophy over the summer.

In addition to looking at student performance for the whole class, we examined performance differences between various subpopulations of the course including: 1) transfer and non-transfer students,[1] 2) students from demographically represented and underrepresented groups in CS in the United States (RG and URG respectively, see Section 4.4 for definitions), and 3) by gender.

We found a difference in performance by gender for one of the three courses, but no clear overall trend. We also found that students from URGs on average do slightly worse on the BDSI than students from RGs for all three courses. However, we find no evidence that this performance gap is widening or closing over time. Our most concerning finding is for the transfer students. Although transfer students perform roughly the same as non-transfer in the first course in the series, they do not improve like their non-transfer peers in the two later courses. This causes a growing knowledge gap between transfer and non-transfer students over time.

These findings are consistent with the recent finding from Alvarado et al. that students who took computing courses prior to college (AP-CS) did better than those who did not, through the entire major [4]. This makes the findings no less problematic. Indeed,

[1] transfer students are generally those who join our university after completing two years of school at a community college

these findings highlight the need for research on interventions capable of offsetting the societal barriers presented to students from lower socioeconomic backgrounds (common among transfer students at our institution) and to offset the lack of access to early computing courses in school districts in the United States with a majority of Black and Latinx students [14].

2 RELATED WORK

It has long been known that students are not learning as much in courses as instructors think they do [10, ch. 2][23]. Moreover, evidence indicates an issue of knowledge retention among students. After an extensive literature review on knowledge retention, Custers found support for about a third of knowledge being lost in the first year, half being lost after a few years, but leveling off subsequently [5]. Tennyson and Beck recently showed that CS2 students who took CS1 in the Fall semester and CS2 in the Spring semester had a knowledge loss of only 4%, whereas students who took CS1 in the Spring semester and CS2 in the Fall semester showed a knowledge loss of 15.1% [24]. This difference may be explained by the long summer break between the Spring and Fall semesters. A study on prerequisite knowledge has shown that upper division CS students enter their courses with substantially less prerequisite knowledge than teachers might expect [27]. However, students did show evidence of improving their prerequisite knowledge during the course that was studied. While these studies all provide great insights, the knowledge retention of a core CS topic for students at various stages in a CS program has not yet been studied.

Considerable work has been done in the space of the recruitment and retention of students from groups that are underrepresented in computing. For example, women in computing have been studied extensively over the years [6–8]. More recently the Computing Education Research (CER) community has begun to recognize and study the underrepresentation of certain minority populations in computing, such as Black and Latinx students in the United States [14].

The gap between the performance of students from represented populations relative to underrepresented populations has been described in STEM education studies, particularly in publications that aim to narrow the achievement gap using active learning techniques [11, 26]. In CS, a recent study showed that adopting Computing in Context [9], Pair Programming [29], and Peer Instruction [15] improved performance for all groups of students with students from URGs benefiting more in absolute numbers, but because the relative rates of improvement were comparable, there was no evidence of the achievement gap narrowing [21].

Transfer students have been studied less extensively, but a study from 2010 presents several recommendations [16]. For example, it is suggested a university should appoint study advisors specifically to advise transfer students in order to provide them with better guidance. But to our knowledge, less is known about the performance of transfer students in computing relative to their non-transfer peers.

There are several assessments available to assess student understanding of CS topics [3, 12, 17–19, 25, 28]. In this paper we chose to use the recently developed Basic Data Structures Inventory (BDSI) [19] because of its rigorous validation process and because we believe data structures is a central topic in virtually any CS program. As such, other institutions should be able to repeat our methods as part of potential replication efforts. Lastly, because basic data structures also appear on industry coding interviews, this is a topic that should be important to students. At the time of writing, the BDSI is not publicly available, but it can be obtained via the instructions appearing in the article by Porter et al. [19].

3 METHODS

This study was conducted at UC San Diego and is approved for Human Subjects Research. The Basic Data Structures Inventory (BDSI), a Concept Inventory (CI) specifically designed to assess student knowledge on fundamental data structures knowledge, was used as our instrument for measuring student proficiency [19].

3.1 Research Questions

We formalized our research questions for the study as follows:

RQ 1: Does general student performance on Basic Data Structures change throughout the various stages of a CS program?

RQ 2: Are there differences in student performance on a Basic Data Structures test for the following subpopulations?
- Transfer students vs non-transfer students
- Men vs women
- Students from URGs vs RGs

RQ 3: Do the performance differences between subpopulations (if any) change across the various stages of a CS program?

3.2 Data Collection

Participants were recruited from the following three CS courses at our university towards the end of the Fall 2020 term. We chose to measure in multiple different courses during the same term instead of following a single cohort because if a student were to complete the BDSI more than once, particularly if answers were discussed at the end of the administration, it would bias later results.

(1) Introduction to Data Structures (CS2)
(2) Introduction to Computer Organization (ORG)
(3) Advanced Data Structures (ADS)

These courses form a prerequisite chain in our department:

$$CS2 \rightarrow ORG \rightarrow ADS$$

CS2 is taken after our introductory CS course (CS1). CS2 and ORG are lower division courses primarily taken by first and second year students. ADS is the gateway course into our upper division which is generally taken by students in their second or third year.

We approached the instructors of these courses to ask for permission to run a study on their students. For most of these courses the content addressed by the BDSI is extremely relevant course prerequisite material. This enabled us to frame our BDSI sessions as prerequisite material review sessions for the final exams.

The instructors informed their students about these sessions. Participation was voluntary and, depending on the course, the students were compensated at the instructor's discretion with extra credit on their final exam or on a programming assignment. Since the BDSI is a Concept Inventory, we cannot give performance based rewards to the students as this would heighten the chances for cheating to occur. Thus students were graded based solely on participation. In order to discourage students from submitting random answers to receive the participation credit without making an honest attempt

	CS2	ORG	ADS
n	221	187	325
mean	5.52	6.14	6.93
std	2.67	2.70	2.92
25%	3	4	5
50%	5	6	7
75%	8	8	9

Table 1: Overall BDSI performance per course. The maximum possible BDSI score is 13.

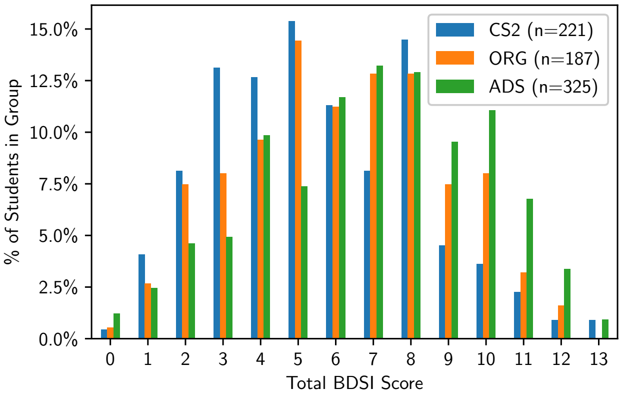


Figure 1: Student BDSI performance plotted per course. A clear shift to the right is visible.

at the BDSI, we told students they would only receive the extra credit if they took at least 30 minutes to complete the test.

For each course, students were given one hour to complete the test. To make it the promised review session (and to provide the students with an additional incentive to take the test seriously), we discussed the answers with the students after they had completed the test and submitted their answers.

We are using the recommended process for BDSI administration as presented in the BDSI paper [19]. However, because of the ongoing COVID-19 pandemic at the time of this study, we were unable to administer the BDSI in person. Instead, we held video conference calls with our participants. The BDSI was provided through a non-downloadable link and responses to the BDSI questions were collected using a Google Form. On the final page of the form we asked students to reflect on the test and provide some personal background information.

4 RESULTS

4.1 BDSI Performance per Course

Table 1 shows the of BDSI score distributions for the three courses. Figure 1 shows a visual overview of the distribution of BDSI performance in CS2, ORG and ADS. An upward trend in later courses is evident, suggesting that students continue to improve their knowledge of basic data structures as they progress through their CS program after completing their CS2 course.

	CS2	ORG	ADS
CS2	1	**0.02**	**2.15e-08**
ORG	**0.02**	1	**0.01**
ADS	**2.15e-08**	**0.01**	1

Table 2: All Students: Holm-Bonferroni-adjusted p-values resulting from a post-hoc Dunn's test for the general student population in each course. Significant values for $\alpha = 0.05$ are bolded. Results indicate significantly different BDSI scores for ADS compared to CS2 and ORG as well as for ORG compared to CS2.

	CS2	ORG	ADS
CS2	1	**0.01**	**2.32e-08**
ORG	**0.01**	1	**0.01**
ADS	**2.32e-08**	**0.01**	1

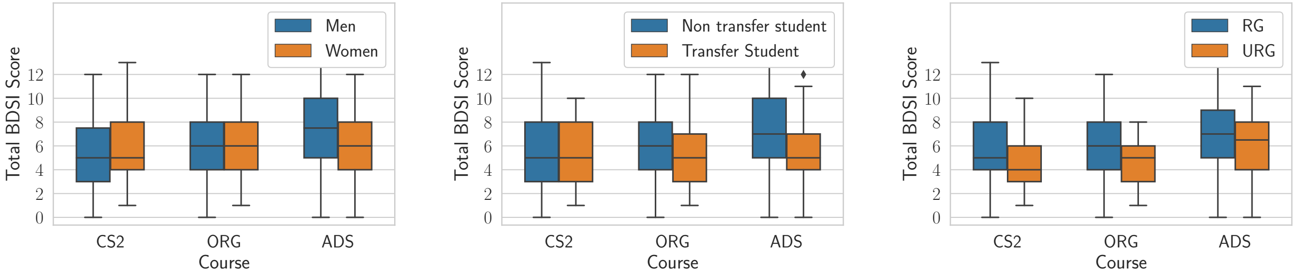
Table 3: Non-Transfer Students: Holm-Bonferroni-adjusted p-values resulting from a post-hoc Dunn's test on non-transfer student BDSI total scores for the three courses. Values indicating significance for $\alpha = 0.05$ are bolded. Results indicate for non-transfer students BDSI score differences between CS2, ORG and ADS are significant.

We determined with a normality test that the BDSI performance data per course is not normally distributed, thus we apply nonparametric tests for our analyses. For example, we use Kruskal–Wallis H tests instead of ANOVAs and Mann-Whitney U tests instead of t-tests. As we are running many statistical tests in this study, we adjust our p-values in order to address the increased chances for Type I errors. We chose to apply the Holm-Bonferroni adjustment method as it is allows for more statistical power and reduces the probability for introduction of Type II errors when compared to the simpler Bonferroni adjustment method [2].

A Kruskal–Wallis H test shows there are significant differences among the BDSI results for the three courses (Kruskal Stat: 34.16, p: 3.83e-08). In order to assess all pairwise differences for significance we ran a post-hoc test for nonparametric data (Dunn's test). As can be seen in Table 2, this test shows that the scores for ADS differ significantly from the scores in both CS2 and ORG and that the scores for ORG differ significantly from the scores for CS2 as well.

4.2 Transfer students

As can be seen in Table 4b, transfer students performed significantly worse than non-transfer students for ORG and ADS. Moreover, Figure 2b shows a clear upward trend for non-transfer students, whereas there is no clear upward trend visible for transfer students. When we run a Kruskal-Wallis H test on the non-transfer students, we find there is a significant difference among the three courses (stat: 33.97, p: 4.19e-08). Following up with a post-hoc Dunn's test (details in Table 3), we find that the scores for non-transfer students significantly improved throughout the entire program. Concerningly, we find no evidence that transfer students improve their performance on the BDSI anywhere from CS2 through ADS (Kruskal-Wallis H stat: 0.52, p: 0.77).



(a) Men and women.

(b) Transfer and non-transfer students. A clear upward trend is visible for the non-transfer students.

(c) RG and URG students.

Figure 2: Box plots of BDSI performance by gender, transfer status and URG status per course.

	CS2		ORG		ADS	
	M	F	M	F	M	F
n	147	71	122	64	210	98
mean	5.41	5.77	6.20	6.11	7.17	6.28
std	2.59	2.83	2.65	2.80	3.01	2.75
25%	3	4	4	4	5	4
50%	5	5	6	6	7.5	6
75%	7.5	8	8	8	10	8
M-W U	stat=4878.5 p=0.43		stat=3777.5 p=0.43		**stat=8415.0** **p=0.01**	

(a) Men and women.

	CS2		ORG		ADS	
	NT	T	NT	T	NT	T
n	160	60	160	26	257	52
mean	5.56	5.45	6.33	5.15	7.16	5.50
std	2.75	2.47	2.66	2.75	2.93	2.68
25%	3	3	4	3	5	4
50%	5	5	6	5	7	5
75%	8	8	8	7	10	7
M-W U	stat=4714.5 p=0.42		**stat=1521.0** **p=0.03**		**stat=4432.5** **p=1.79e-04**	

(b) Transfer and non-transfer students.

	CS2		ORG		ADS	
	RG	URG	RG	URG	RG	URG
n	199	22	169	17	282	28
mean	5.64	4.45	6.30	4.88	6.98	6.14
std	2.66	2.50	2.72	2.11	2.96	2.75
25%	4	3	4	3	5	4
50%	5	4	6	5	7	6.5
75%	8	6	8	6	9	8
M-W U	stat=1621.5 p=0.06		stat=998 p=0.06		stat=3313.0 p=0.08	

(c) RG and URG students.

Table 4: BDSI performance distribution details per course and subpopulation corresponding to the box plots in Figure 2. Statistically significant differences between subpopulations are bolded (for $\alpha = 0.05$ according to a Holm-Bonferroni-adjusted Mann-Whitney U test).

4.2.1 *Could it be the case that transfer students who took CS2 at our institution do better than those who took a basic data structures course elsewhere?*

We asked ourselves this question after we noticed that the BDSI performance differences between transfer students and non-transfer students at the end of our CS2 course were not statistically significant. Thus we suspected that perhaps what mattered most for the difference in performance between transfer students and non-transfer students in the following courses was not caused by whether they were transfer students, but whether they had taken the same CS2 course.

However, contrary to our expectations, the transfer students in ADS who took CS2 at a different institution performed *better* on the BDSI than transfer students who had taken CS2 at our institution. This difference, though, was not statistically significant. Figure 3 and Table 5 show the details of the distributions. Students who did not report their CS2 institution were excluded from the analysis.

4.3 Gender

The differences in performance between men and women are not significant except in ADS, where we find men performed significantly better on the BDSI. Details about the performance distributions can be seen in Figure 2a and Table 4a. Section 5.2.1 discusses a possible explanation for why we may suddenly see this difference.

4.4 Underrepresented Groups

UC San Diego defines students from represented groups in CS (RG) as those who are Caucasian (non Hispanic or Latinx) and Asian.

	CS2 at our institution	CS2 not at our institution
n	27	17
mean	5.26	6.47
std	2.73	2.45
25%	3.5	5
50%	5	6
75%	7	8
	Mann-Whitney U stat=170.0, p=0.08	

Table 5: BDSI performance distribution of transfer students in ADS who took CS2 at our institution and those who took CS2 at a different institution. The difference is not statistically significant for $\alpha = 0.05$.

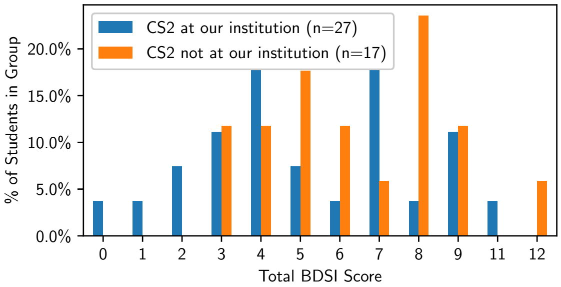


Figure 3: BDSI performance of transfer students in ADS who took CS2 at our institution plotted against those who took CS2 at a different institution.

	CS2	ORG	ADS
CS2	1	**0.03**	**6.72e-07**
ORG	**0.03**	1	**0.03**
ADS	**6.72e-07**	**0.03**	1

Table 6: RG Students: Holm-Bonferroni-adjusted p-values resulting from a post-hoc Dunn's test for the BDSI performance of RG students. Significant values for $\alpha = 0.05$ are bolded. Results show that for RG students BDSI scores differ significantly between all three courses.

Students from underrepresented groups in CS (URG) are defined as those who are Black, Hispanic or Latinx, Native American and Pacific Islander. Students who did not report their demographic background to the university were not included in the results pertaining to RGs and URGs. We find that, on average, RG students performed better than URG students in all courses. However, we were not able to establish statistical significance for these differences. Details can be found in Figure 2c and Table 4c.

A Kruskal-Wallis H test shows that for RG students there is at least one strongly significant improvement from CS2 through ADS (26.99, p: 1.38e-06). A post-hoc Dunn's test (see Table 6) shows significant differences between the BDSI scores for all three courses for RG students.

We see slightly different results for students from URGs. Although, judging by the average scores, URGs do seem to increase their BDSI performance throughout CS2, ORG and ADS, and a Kruskal–Wallis H test shows near significance (5.80, p: 0.055), we did not find any significant differences between courses in a post-hoc Dunn's test.

5 DISCUSSION

Our results led to several interesting findings that we will discuss in the context of our research questions.

5.1 RQ 1: BDSI Performance at Different Stages in Our CS Program

We find that general student performance on the BDSI improved throughout the progression of courses in our CS program. For example, the median score increased consistently from CS2 through ADS. Moreover, we find that the performance increases between the courses are statistically significant. These findings are encouraging as they imply students continue to improve their understanding of CS fundamentals as they progress through our program. However, we have some concern whether a 7 out of 13 in ADS represents mastery of the material. Unfortunately, the BDSI is a relatively new instrument and we know of no articles on student performance later in their CS careers against which we might compare.

5.2 RQ 2: Differences in Performance Between Subpopulations

5.2.1 Gender. We encouragingly find that men and women mostly perform equally well on the BDSI, despite the many sociocultural barriers that cause women to be less likely to gain access to computing education prior to university in our country [1, 7]. Only for ADS do we find a statistically significant difference in performance. However, we suspect this may be due to the fact that the women in our courses are less likely than the men to be CS majors (in ADS 69% of men were CS majors vs 49% of women). At our institution, other majors are allowed, at times, to bypass ORG to reach ADS and our department has found that those who take ADS without ORG are less likely to do well.

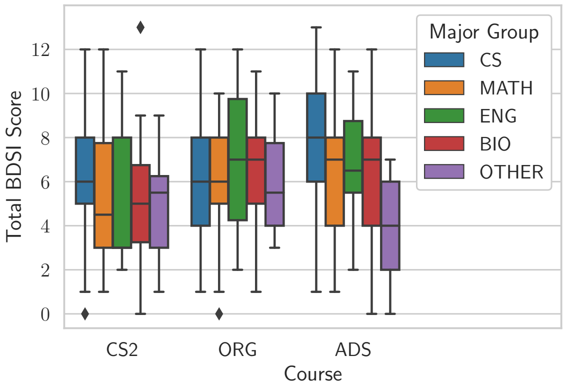


Figure 4: Student BDSI performance plotted per course and major group.

Indeed, we see in Figure 4 for ADS that CS majors scored substantially higher than other majors, whereas the differences between majors were not as substantial in the other courses. In ADS, 49% of women were Math and Biology-related majors, which both scored substantially lower on average than CS majors. As only 27% of men in ADS were Math and Biology-related majors, this could potentially explain why we are seeing a performance difference between men and women in ADS. Moreover, we encouragingly find no evidence that the performance difference between men and women is significant *within CS majors* in ADS.

5.2.2 Transfer Students. We find that transfer students perform significantly worse on the BDSI than non-transfer students in all courses except CS2. This concerning finding prompted us to investigate whether transfer students who took CS2 at our institution were at an advantage compared to their transfer student peers who had taken CS2 elsewhere. However, contrary to our expectations, transfer students who took CS2 at our institution do not seem to have an advantage over transfer students who took CS2 elsewhere. In fact, our data seems to suggest the opposite may be true. A possible explanation is that most of the community colleges that our transfer students attend are on a semester system, and we are on a quarter system. It is possible that the additional 4–5 weeks of content might explain the difference. However, we remain concerned that overall transfer students still perform worse than non-transfer students in our courses, and call for future research to determine the cause.

5.2.3 Underrepresented Groups. We find that students from URGs consistently score lower on average than students from RGs. However, the combination of small sample sizes of URGs and the small differences with RGs makes it difficult to attach any statistical significance to the observed differences. When we filter out transfer students and compare only non-transfer RGs and non-transfer URGs, the performance gap widens further.

Interestingly, in our preliminary investigations using self-reported URG status, we found much stronger effects than in our analysis on the official URG status data as provided by our institution. This suggests there may be merit in investigating the power of self-reported URG status when compared to an automatically assigned demographics-based status, especially in light of recent work on the impostor phenomenon and the importance of sense of belonging [20, 22].

5.3 RQ 3: Do Performance Differences Change Over Time?

We investigated whether the performance differences between the subpopulations change or remain the same throughout the different stages in the program. Our findings for this question are concerning.

5.3.1 Transfer students. For transfer students the performance gap compared to non-transfer students appears at all stages of the program that we measured. Moreover, the issue worsens over time, with the gap widening with each succeeding course. While non-transfer students significantly improve their understanding of data structures, transfer students appear to remain at the same level of proficiency as they progress through the program. This finding is deserving of future inquiry to see if it is consistent across multiple institutions and to determine potential sources of the problem.

5.3.2 Underrepresented Groups. Similarly, the performance difference between students from URGs and RGs appears in all three courses. Moreover, while we find that RGs improve significantly over time, we do not find statistically significant improvements between courses for URGs. However, we observe no clear trend that the difference between RGs and URGs increases or decreases over time.

We know from prior work that, compared to RGs, students from URGs have less exposure to computing before attending university [14]. We also know from prior work that prior experience (or lack thereof) causes performance gaps in CS courses throughout the entire curriculum [4]. Based on these findings one might wonder whether lack of prior experience causes a persisting gap in understanding of basic data structures concepts even for those who persevere in the major. This, too, is deserving of future inquiry.

6 LIMITATIONS AND THREATS

The students in our dataset all attend UC San Diego; our results may not generalize beyond that context. One limitation of our findings is that we did not measure performance over time for the same population of students. This would be difficult. For the "review sessions" recommended by the BDSI creators [19], the benefit to students is the subsequent discussion of the correct answers. Also, there is only one version of the BDSI. These factors limit our ability to administer the same exam to the same population more than once. We have assumed the populations of students are similar for our assessed courses, and believe this to be reasonable based on our institutional knowledge. However, a threat to validity remains that the populations of students are different.

The BDSI instrument we used assesses just one subject that has a unique position in the typical CS curriculum. The results may not generalize to other subjects. Furthermore, working in a field of practice leads students to constantly advance their knowledge by themselves (e.g., studying for interviews), so good performance on the BDSI cannot necessarily be attributed to what was taught nor the quality of instruction.

A threat to external validity is that the BDSI was originally designed for in-person sessions, whereas, due to the ongoing COVID-19 pandemic, we administered the BDSI over video calls. This may have had an impact on student performance, so our results may not be comparable to studies where the BDSI was administered in person. We attempted to minimize this risk by telling students to treat the online BDSI session like a test on paper in a lecture hall, and by asking them to not discuss or search for answers online.

Future work could explore a wider array of institutions, graduating students, and evaluate knowledge on topics beyond basic data structures. Access to student grades in CS2 could generate useful additional insight. Likewise, learning students' interview preparation strategies could shed further light on the sources of graduating student knowledge.

7 CONCLUSION

This study investigated the basic data structures proficiency of several student subpopulations at various stages in a CS program. Encouragingly, we find that the average student proficiency with basic data structures increases for courses later in the CS program. This evidence suggests that students, on average, continue to learn core concepts like basic data structures as they progress through the program.

Concerningly, we find that transfer students perform significantly worse than non-transfer students in both ORG and ADS. Furthermore, we find that this performance gap widens as students progress further in their studies. Similarly to transfer students, students from URGs in CS do not perform as well on the BDSI as students from RGs, but this difference is not statistically significant and we do not find any evidence for a widening or closing of this performance gap over time. The performance for both transfer students and students from URGs (despite a visible trend in Figure 2c for URGs) does not statistically significantly improve in courses over time, whereas performance of non-transfer students and students from RGs increased significantly.

These findings are concerning for the diversity, equity, and inclusion of computing courses, particularly in light of the socioeconomic diversity of transfer students. It is possible that the systemic societal problems in our country that limit access to computing before college for students with lower socioeconomic status (often transfer students) and those from URGs [14] continues to impact student performance even after many courses in our major. This unsettling finding would be consistent with the work of Alvarado et al. who found that students who took CS in high school outperform those who did not, all the way through the CS curriculum [4]. We believe these findings are a call to action for more research into the sources and extent of the problem as well as possible interventions that aim to close the performance gap.

8 ACKNOWLEDGEMENTS

We thank instructors Greg Miranda, Bryan Chin, Niema Moshiri and Miles Jones for agreeing to let us run this study in their courses. This work was supported in part by NSF award DUE-1712508. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant

No. DGE-1650112. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Statistics for Advanced Placement (AP) Computer Science Students in High Schools. https://code.org/promote/ap

[2] Mikel Aickin and Helen Gensler. 1996. Adjusting for multiple testing when reporting research results: the Bonferroni vs Holm methods. *American journal of public health* 86, 5 (1996), 726–728.

[3] Vicki L. Almstrum, Peter B. Henderson, Valerie Harvey, Cinda Heeren, William Marion, Charles Riedesel, Leen-Kiat Soh, and Allison Elliott Tew. 2006. Concept Inventories in Computer Science for the Topic Discrete Mathematics. *SIGCSE Bull.* 38, 4 (June 2006), 132–145. https://doi.org/10.1145/1189136.1189182

[4] Christine Alvarado, Gustavo Umbelino, and Mia Minnes. 2018. The persistent effect of pre-college computing experience on college CS course grades. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education.* 876–881.

[5] Eugène J.F.M. Custers. 2010. Long-term retention of basic science knowledge: A review study. *Advances in Health Sciences Education* 15, 1 (2010), 109–128.

[6] Benjamin J. Drury, John Oliver Siy, and Sapna Cheryan. 2011. When Do Female Role Models Benefit Women? The Importance of Differentiating Recruitment From Retention in STEM. *Psychological Inquiry* 22, 4 (2011).

[7] Allan Fisher and Jane Margolis. 2002. Unlocking the Clubhouse: The Carnegie Mellon Experience. *SIGCSE Bull.* 34, 2 (June 2002), 79–83. https://doi.org/10.1145/543812.543836

[8] Karen A. Frenkel. 1990. Women and Computing. *Commun. ACM* 33, 11 (Nov. 1990), 34–46. https://doi.org/10.1145/92755.92756

[9] Mark Guzdial. 2013. Exploring hypotheses about media computation. In *Proceedings of the ninth annual international ACM conference on International computing education research.* 19–26.

[10] Mark Guzdial. 2015. Learner-Centered Design of Computing Education: Research on Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics* 8 (11 2015), 1–165. https://doi.org/10.2200/S00684ED1V01Y201511HCI033

[11] David C Haak, Janneke HilleRisLambers, Emile Pitre, and Scott Freeman. 2011. Increased structure and active learning reduce the achievement gap in introductory biology. *Science* 332, 6034 (2011), 1213–1216.

[12] Geoffrey L. Herman, Michael C. Loui, and Craig Zilles. 2011. Students' Misconceptions About Medium-Scale Integrated Circuits. *IEEE Trans. on Educ.* 54, 4 (Nov. 2011), 637–645. https://doi.org/10.1109/TE.2011.2104361

[13] Sophia Krause-Levy, Sander Valstar, Leo Porter, and William G. Griswold. 2020. Exploring the Link Between Prerequisites and Performance in Advanced Data Structures. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20).* Association for Computing Machinery, New York, NY, USA, 386–392. https://doi.org/10.1145/3328778.3366867

[14] Jane Margolis, Rachel Estrella, Joanna Goode, Jennifer Jellison Holme, and Kim Nao. 2017. *Stuck in the shallow end: Education, race, and computing.* MIT press.

[15] Eric Mazur. 1997. Peer instruction: Getting students to think in class. *AIP Conference Proceedings* 399 (1997).

[16] Karen Owens. 2010. Community College Transfer Students' Adjustment to a Four-Year Institution: A Qualitative Analysis. *Journal of The First-Year Experience & Students in Transition* 22, 1 (2010), 87–128.

[17] Miranda C Parker, Mark Guzdial, and Shelly Engleman. 2016. Replication, validation, and use of a language independent CS1 knowledge assessment. In *Proceedings of the 2016 ACM conference on international computing education research.* 93–101.

[18] Leo Porter, Saturnino Garcia, Hung-Wei Tseng, and Daniel Zingaro. 2013. Evaluating Student Understanding of Core Concepts in Computer Architecture. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '13).* Association for Computing Machinery, New York, NY, USA, 279–284. https://doi.org/10.1145/2462476.2462490

[19] Leo Porter, Daniel Zingaro, Soohyun Nam Liao, Cynthia Taylor, Kevin C. Webb, Cynthia Lee, and Michael Clancy. 2019. BDSI: A Validated Concept Inventory for Basic Data Structures. In *Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER '19).* Association for Computing Machinery, New York, NY, USA, 111–119. https://doi.org/10.1145/3291279.3339404

[20] Adam Rosenstein, Aishma Raghu, and Leo Porter. 2020. Identifying the Prevalence of the Impostor Phenomenon Among Computer Science Students *(SIGCSE '20).*

[21] Adrian Salguero, Julian McAuley, Beth Simon, and Leo Porter. 2020. A Longitudinal Evaluation of a Best Practices CS1 *(ICER '20).*

[22] Linda J. Sax, Jennifer M. Blaney, Kathleen J. Lehman, Sarah L. Rodriguez, Kari L. George, and Christina Zavala. 2018. Sense of Belonging in Computing: The Role of Introductory Courses for Women and Underrepresented Minority Students. *Social Sciences* 7, 8 (2018). https://doi.org/10.3390/socsci7080122

[23] Elliot Soloway. 1986. Learning to program = learning to construct mechanisms and explanations. *Commun. ACM* 29, 9 (1986), 850–858.

[24] Matthew F. Tennyson and Marc Beck. 2018. A Study of Knowledge Retention in Introductory Programming Courses. *J. Comput. Sci. Coll.* 33, 4 (April 2018), 13–20.

[25] Allison Elliott Tew and Mark Guzdial. 2011. The FCS1: A Language Independent Assessment of CS1 Knowledge. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11).* Association for Computing Machinery, New York, NY, USA, 111–116. https://doi.org/10.1145/1953163.1953200

[26] Elli J Theobald, Mariah J Hill, Elisa Tran, Sweta Agrawal, E Nicole Arroyo, Shawn Behling, Nyasha Chambwe, Dianne Laboy Cintrón, Jacob D Cooper, Gideon Dunster, et al. 2020. Active learning narrows achievement gaps for underrepresented students in undergraduate science, technology, engineering, and math. *Proceedings of the National Academy of Sciences* 117, 12 (2020), 6476–6483.

[27] Sander Valstar, William G. Griswold, and Leo Porter. 2019. The Relationship between Prerequisite Proficiency and Student Performance in an Upper-Division Computing Course. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19).* Association for Computing Machinery, New York, NY, USA, 794–800. https://doi.org/10.1145/3287324.3287419

[28] Kevin C. Webb and Cynthia Taylor. 2014. Developing a Pre- and Post-Course Concept Inventory to Gauge Operating Systems Learning. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14).* Association for Computing Machinery, New York, NY, USA, 103–108. https://doi.org/10.1145/2538862.2538886

[29] Linda L Werner, Brian Hanks, and Charlie McDowell. 2004. Pair-programming helps female computer science students. *Journal on Educational Resources in Computing (JERIC)* 4, 1 (2004), 4–es.