



Lab 6: Introduction to Hashing: Maps, Sets, Unordered Sets and Maps

Instructions:

Work on this document with your group, then enter the answers on the canvas quiz.

Note:

The only portions of this assignment that will be graded are the feedback form and the hand-written problem. All additional questions are optional and will not be graded.

1 Part A: Feedback Form (15 points)

This is one of the two parts of this lab assignment that will be graded for points. The handwritten problem (part D) will also be graded for points. All other parts are optional and will not be graded (however, feel free to use them as practice!).

Congratulations on finishing the first half of the course! For this week's lab assignment, please complete the following feedback form: <https://forms.gle/djraTvV3AGdGxLh29>. Completion of this survey is needed to earn these points. The survey is due Monday, October 28, at 11:59 pm.

This feedback form serves two purposes.

It helps us gather and analyze data that we can use to improve 281 in the future.

It serves as a structured reflection for you to assess your personal progress and status in 281.

The survey is very detailed. You should plan to spend at least 30 minutes on it. While it may seem long and detailed, there is a method to our madness. The questions serve as a form of directed reflection, so that you can identify specific portions of the course you could improve in, and then come up with techniques to do so. Take it seriously, be honest, and most importantly, take something away from the survey and use it to do even better the rest of the semester.

Furthermore, please take some time to provide feedback for your lab instructor as well! This isn't required as part of the 15 points, but your help will be greatly appreciated. The form is anonymous, and the information you provide will allow the instructors to make any needed adjustments to improve your learning environment. The form can be found here: <https://forms.gle/Kfk4TB8DbCSEaDfN6>

2 Part B: Logistics (UNGRADED)

1. How is this lab graded?
 - A. If 75% of the students fill out the Midterm Feedback form, then everyone gets points.**
 - B. Nothing is graded
2. Is the handwritten problem graded in this lab?
 - A. Yes**
 - B. No
3. What is the due date of Project 3?
 - A. October 31, 2024
 - B. November 12, 2024**
 - C. November 7, 2024
 - D. November 10, 2024
4. Which of the following are covered in this lab?
 - A. Maps**
 - B. Sets**
 - C. Enum Classes**
 - D. Emplacement**
 - E. Using Declarations**

3 Part C: Hash Tables (UNGRADED)

1. For which of the following applications would a hash table NOT be appropriate? Select all that apply.

- A. printing out all of the elements in sorted order**
- B. storing passwords that can be looked up by username
- C. returning a person's name given their phone number
- D. finding the k^{th} largest element in an array**
- E. creating an index for an online book

2. Two pairs (a, b) and (c, d) are symmetric if $b = c$ and $a = d$. Suppose you are given an array of pairs, and you want to find all symmetric pairs in the array. The first element of all pairs is distinct.

For example, if you were given the following array:

```
arr1[] = {(14, 23), (11, 2), (52, 83), (49, 38), (38, 49), (2, 11)};
```

you would print out the symmetric pairs $\{(11, 2), (2, 11)\}$ and $\{(49, 38), (38, 49)\}$. What is the average-case time complexity of accomplishing this task if you use the most efficient algorithm? If hashing is involved, assume that both search and insert methods work in $\Theta(1)$ time.

- A. $\Theta(1)$
 - B. $\Theta(\log n)$
 - C. $\Theta(n)$**
 - D. $\Theta(n \log n)$
 - E. $\Theta(n^2)$
3. Given two unsorted arrays of distinct integers, you want to find all pairs of numbers, one from array 1 and one from array 2, that sum to a given value. For instance, if you are given

```
arr1[] = {1, 2, 3, 4, 5, 7, 11}  
arr2[] = {2, 3, 4, 5, 6, 8, 12}
```

you would return (1, 8), (3, 6), (4, 5), (5, 4), and (7, 2). What is the average time complexity of doing this if you use the most efficient algorithm?

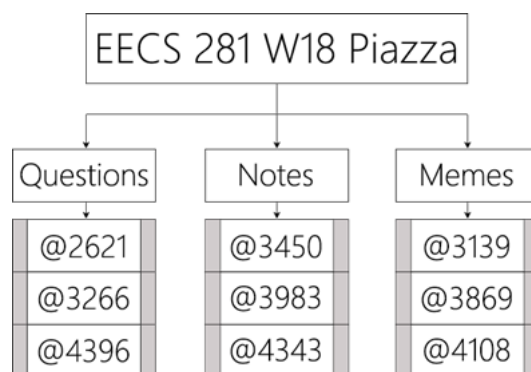
- A. $\Theta(1)$
 - B. $\Theta(\log n)$
 - C. $\Theta(n)$**
 - D. $\Theta(n \log n)$
 - E. $\Theta(n^2)$
4. Which of the following statements is false?
- A. the `std::map` data structure is implemented using a binary search tree under the hood
 - B. unlike a `std::set`, a `std::map` has a value associated with each key

- C. using `operator[]` on a nonexistent key in a `std::unordered_map` causes the key to exist
- D. searching for an element in a hash table is guaranteed to always take constant $\Theta(1)$ time.**
- E. none of the above
5. You are given an array of n elements where elements may be repeated, and you want to find the minimum number of elements that need to be deleted from the array so that all elements in the array are equal. For instance, if you are given the following array
- ```
arr1[] = {3, 6, 8, 6, 2, 7, 6, 3, 1, 3, 6}
```
- you would return 7, as this is the minimum number of deletions required to obtain an array where all elements are the same (in this case, you would get an array with all 6's). What is the average time complexity of doing this if you use the most efficient algorithm?
- A.  $\Theta(1)$
- B.  $\Theta(\log n)$
- C.  $\Theta(n)$**
- D.  $\Theta(n \log n)$
- E.  $\Theta(n^2)$
6. The posts on the Spring 2018 EECS 281 Piazza page can be categorized into three unique groups: questions, notes, and memes.

Suppose you implemented the following unordered map that maps the type of each post to the post IDs that correspond to each of these types (for instance, the key `notes` maps to 3450,3983,4343):

```
std::unordered_map<std::string, std::vector<int>> posts;
```

While searching through Piazza, you discover that post @4753 is a meme. Which of the following successfully appends 4753 to the vector associated with the key `memes`?



- A. `posts["memes"] = 4753;`
- B. `posts["memes"] = memes.push_back(4753);`
- C. `posts["memes"] = posts.push_back(4753);`
- D. `posts["memes"].push_back(4753);`**
- E. `posts["memes"].second.push_back(4753);`

7. Consider the following snippet of code:

```
int main() {
 unordered_map<string, string> profs;
 profs.insert(make_pair("Paoletti", "Darden"));
 profs.emplace("Garcia", "Darden");
 profs.insert(make_pair("Paoletti", "Garcia"));
 profs["Garcia"] = "Paoletti";
 profs.insert(make_pair("Paoletti", "Markov"));
 cout << profs["Paoletti"] << endl;
 cout << profs["Darden"] << endl;
 profs.erase("Paoletti");
 cout << profs["Garcia"] << endl;
 cout << profs.size() << endl; // What gets printed here?
}
```

What is the size of profs after all operations are completed? In other words, what does the last line of code print?

- A. 1
- B. 2**
- C. 3
- D. 4
- E. 7

8. Consider the following snippet of code:

```
int main() {
 map<int, pair<int, int>> elts;
 elts[14] = make_pair(12, 13);
 elts[17] = make_pair(10, 15);
 elts[11] = make_pair(16, 19);
 elts[18] = make_pair(14, 12);
 elts[13] = make_pair(15, 17);
 elts[12] = make_pair(11, 10);
 vector<int> vec;
 for (auto myVar : elts)
 vec.push_back(myVar.first);
 cout << vec.front() << endl; // What gets printed here?
 return 0;
}
```

What does this code print?

- A. 10
- B. 11**
- C. 12
- D. 14
- E. 16

9. The Project 3 autograder just got released! On the first day, seven students submitted. The scores that each of these students received on their first submission are shown in the table below.

| Name  | Alice | Bob  | Cathy | Drew | Erin | Frank | Grace | Henry |
|-------|-------|------|-------|------|------|-------|-------|-------|
| Score | 34.9  | 46.0 | 14.6  | 65.1 | 42.1 | 28.0  | 96.7  | 74.5  |

Suppose these students are placed into a map named `P4Scores`, where `Name` represents the key and `Score` represents the value. If `it = P4Scores.end()`, what is the value of `(--it)->first`?

- A. Drew
  - B. Cathy
  - C. Frank
  - D. Grace
  - E. Henry**
10. Which of the following is not a characteristic of a good hash function?
- A. the capability to distribute keys evenly in a hash table
  - B. the capability to cluster similar keys together in a hash table**
  - C. the capability to compute a hash for every possible key
  - D. the capability to compute the same hash for the same key
  - E. none of the above (i.e. all of the above are characteristics of a good hash function)
11. A hash table of size 100 has 40 empty elements and 25 ghost elements. What is its load factor?
- A. 0.25
  - B. 0.35**
  - C. 0.60
  - D. 0.65
  - E. 0.75
12. You want to write a hash function that will distribute keys over 10 buckets, numbered 0 to 9, for integer keys  $i$  ranging from 0 to 2019. The hash table is implemented using separate chaining. Of the following hash functions, which one would be best at distributing the keys uniformly over all 10 buckets? *Hint: Which three options can you eliminate immediately? Of the two that remain, write out a few hash values and see if you can identify a pattern.*
- A.  $H(i) = 3i^2 \bmod 10$
  - B.  $H(i) = 6i^2 \bmod 10$
  - C.  $H(i) = 9i^3 \bmod 10$**
  - D.  $H(i) = 12i^3 \bmod 10$
  - E.  $H(i) = 15i^4 \bmod 10$

## 4 Part D: Handwritten Problem

---

13. Given an array of distinct integers, find if there are two pairs (a, b) and (c, d) such that  $a+b=c+d$ , and a, b, c, and d are distinct elements. If there are multiple elements, the function should print all pairs that work. You may assume that for any pair (a, b), there is at most one other pair (c, d) that sums to  $a+b$ .

**Example:**

Input: [3, 4, 7, 1, 2, 8]

Output:

```
(3, 2) and (4, 1) //5
(3, 7) and (2, 8) //10
(3, 8) and (4, 7) //11
(7, 2) and (1, 8) //9
```

Input: [65, 30, 7, 90, 1, 9, 8]

Output(nothing):

```
// Prints out all different pairs in input_vec that have same sum.
// Expected Runtime: $O(n^2)$
void two_pair_sums(const vector<int> &input_vec, ostream& out);
```