Week 4: Wednesday

EECS 281

MAXIM ALEKSA

maximal.io

Walking the halls of BBB late at night, Dr. P came upon a student in distress. A deadline was fast approaching, and after three all-nighters the exhausted student had dropped his only remaining coins into the soda machine without first noticing the paper sign taped to it.

The sign read `OUT OF ORDER` in large red capitals.

“Can you fix it?” asked the student, who knew of Paoletti’s skill with primitive machines.

Dr. P studied the machine for a moment, then crossed out the letters of the sign and under them wrote `DEFOOORRTU`. Immediately a soda can dropped into the chute.

Satisfied, Dr. P continued on his way.

(Adapted from UC Berkeley)

(Non-exhaustive) list of topics

Asymptotics. Analysis of algorithms. Recurrence relations.

STL. Iterators. Containers. Algorithms.

Arrays. Linked Lists. Stacks. Queues.

Priority Queues. Heaps.

Sorting Algorithms. Bubble sort. Selection sort. Insertion sort. Heapsort. Merge sort. Quicksort

Sorting properties

Computational complexity: best case, average case, worst case. Comparison-based algorithms need at least O(n log n) comparisons on most inputs.

Memory: A sort is in-place if it uses a small, constant amount of additional memory.

Adaptiveness: does the pre-sortedness of the input positively affects the running time?

Stability: maintain the relative order of equal items.

Find Sum

Interview Question

Given an integer **`sum`** and a sorted array **`numbers`** of **`N`** distinct integers, implement a function to find if there exist indices **`i`** and **`j`** such that **`numbers[i] + numbers[j] == x`**.

```
bool findSum(const vector<int> &numbers, int sum) {
    for (int i = 0; i < numbers.size(); i += 1) {
        for (int j = 0; j < numbers.size(); j += 1) {
            if (numbers[i] + numbers[j] == sum) {
                return true;
            }
        }
    }
    return false;
}
```

Find Sum

Interview Question

Given an integer **`sum`** and a sorted array **`numbers`** of **`N`** distinct integers, implement a function to find if there exist indices **`i`** and **`j`** such that **`numbers[i] + numbers[j] == x`**.

```
bool betterFindSum(const vector<int> &numbers, int sum);
```

Implement a better, more efficient version of the algorithm.