

**Due:** N/A. You do **not** need to hand this in

**Worth:** 0%. Not marked

These are some sample questions from materials you should already know. Some of the questions are harder and/or require some creativity; these are marked with a (\*). If you struggle with a lot of these consider putting a lot of time into learning the prerequisites; you can show up at the office hours for pointers.

*Be warned: If you are having a lot of problems with these questions, specially the ones near the beginning and are not willing to put in the extra time you will run into trouble later on in the course. These are some of the things I will assume you already know!*

1. Mark each of the following as true/false.

$n = O(n^2)$	$n = O(n)$	$n = \Omega(n^2)$	$n = o(n)$
$n = o(2^n)$	$n^2 = \Omega(2^n)$	$2^n = o(3^n)$	$\log_3 n = o(\log_2 n)$
$\log(n!) = o(n \log n)$	$10n = O(n)$	$2^{10n} = O(2^n)$	$\log(10n) = O(\log n)$
$\log(n) = o(\log(n^2))$	$\log(n \log n) = O(\log n)$	$\log \log 2^n = O(\log \log n)$	$\log 3^n = O(\log 2^n)$
(*)	$2^n = o(2^{n+1})$	$2^{2^n} = o(2^{2^{n+1}})$	$\log^* n = O(\log \log \log n)$

2. Consider a function  $T(n)$  defined recursively. For each of the following definitions give the asymptotic solution.

a)	$T(1) = 1$	$T(n) = 10T(n/5) + \Theta(n)$
b)	$T(1) = 1$	$T(n) = 3T(n/3) + \Theta(n^2)$
c)	$T(1) = 1$	$T(n) = 4T(n/3) + \Theta(n^{\log_3 4})$
d)	$T(1) = 1$	$T(n) = T(n/2) + \Theta(n)$
e)	$T(1) = 1$	$T(n) = T(\sqrt{n}) + 1$
f)	$T(1) = 1$	$T(n) = T(n-1) + \log n$
g)(*)	$T(0) = 1, T(1) = 2$	$T(n) = 2T(n-1) + 3T(n-2)$

3. Consider a function  $T(n)$  defined recursively. For each of the following definitions give the precise solution.

g)	$T(1) = T(2) = 1$	$T(n) = T(n-1) + T(n-2)$
b)	$T(1) = 1$	$T(n) = T(n-1) + n$
c)	$T(1) = 1$	$T(n) = T(n-1) + 1/n$
g)(**)	$T(0) = 1, T(1) = 2$	$T(n) = 2T(n-1) + 3T(n-2)$

4. Assume that as input you are given a sequence of  $n$  numbers  $A[0], A[1], \dots, A[n-1]$ . Further, assume that the sequence is concave, i.e. for all  $0 \leq i \leq j \leq k < n$ ,  $(k-i)A[j] \geq (k-j)A[i] + (j-i)A[k]$ . The goal is to find the maximum of the sequence, i.e.  $\max_i A[i]$ . Consider two solutions presented in Algorithms 1 and 2.

- (a) Show that Algorithms 1 and 2 are correct.
- (b) Compute the asymptotic runtime of Algorithms 1 and 2 are correct.

---

**Algorithm 1:** Linear Search

---

**Input:** Integer  $n$ , Concave sequence  $A[0], \dots, A[n-1]$ .**Output:**  $\max_i A[i]$ .

```
1  $x \leftarrow -\infty$ 
2 for  $i \leftarrow 0$  to  $n - 1$  do
3    $x \leftarrow \max(x, A[i])$ 
4 end
5 return  $x$ 
```

---

---

**Algorithm 2:** Search

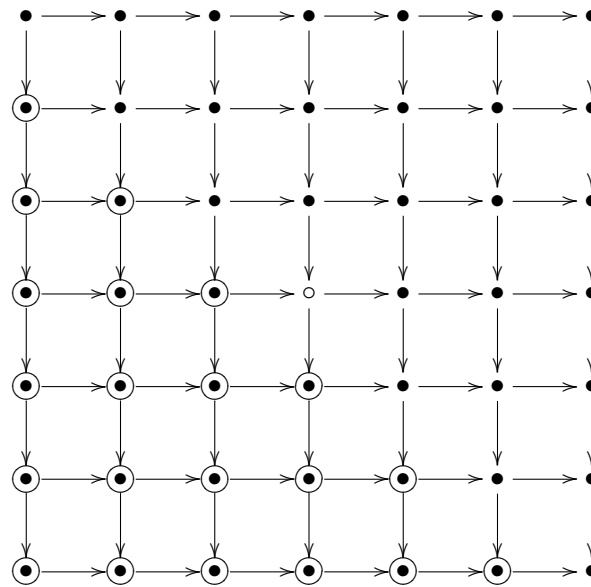
---

**Input:** Integer  $n$ , Concave sequence  $A[0], \dots, A[n-1]$ .**Output:**  $\max_i A[i]$ .

```
1 begin
2   RSearch( $A, 0, n$ )
3 end
4 RSearch( $A, from, to$ )
5 begin
6   if  $from + 1 == to$  then
7     return  $A[from]$ 
8   end
9    $mid \leftarrow \lfloor (from + to) / 2 \rfloor$ 
10  if  $A[mid] > A[mid - 1]$  then
11    return RSearch( $A, mid, to$ )
12  else
13    return RSearch( $A, from, mid$ )
14  end
15 end
```

---

5. In each of the following problems you receive a sequence of *commands* and have to keep track of a *working set* of numbers. How do you do that; in particular, which data structure would you use? You can always assume that all commands are valid and you don't need to actually store the working set; all you need to do is being able to process the commands, i.e. answer the ones that require you to print something.
- The possible commands are "INS  $x$ ", "DEL  $x$ ", and "SEARCH  $x$ ", for inserting a number in the working set, deleting a number from the working set and printing yes/no based on whether  $x$  is in the working set.
  - The possible commands are "INS  $x$ " and "SEARCH  $x$ ", for inserting a number in the working set, and printing yes/no based on whether  $x$  is in the working set.
  - The possible commands are "INS  $x$ " and "SELECT  $k$ ", for inserting a number in the working set, and printing the  $k$ th smallest number in the working set.
  - The possible commands are "INS  $x$ ", "DEL  $x$ " and "SUM". For inserting a number in the working set, deleting a number from the working set and printing the sum of the numbers currently in the working set.
  - The possible commands are "INS  $x$ ", "DELMIN" and "MIN", for inserting a number in the working set, deleting the smallest number in the working set and printing the smallest number in the working set.
6. The following question has to do with a graph. Like the last question you will get a sequence of commands you have to process while keeping track of a graph. The first command is always "SIZE  $n$ ". You have to process the following commands:
- "SIZE  $n$ " Initialize a graph  $G$  with  $n$  vertices  $v_1, \dots, v_n$  and no edges. This is always the first command and you will only get it only once.
- "ADDE  $i, j$ " Add an edge between the vertices  $v_i$  and  $v_j$  in your graph.
- "CONNECTED  $i, j$ ?" Print yes if the vertices  $v_i$  and  $v_j$  are connected in the current graph.
- For example for the sequence "SIZE 10; CONNECTED 1, 2? ADDE 1, 2; ADDE 1, 3; CONNECTED 2, 3?; ADDE 3, 4; CONNECTED 2, 5?; CONNECTED 2, 4?" you should output "no, yes, no, yes".
7. They are 80 students enrolled in CSC373 this term (this is true) and the class is full. There are going to be two tutorial sections how many ways are there to assign the students to the tutorials? What if we require that the sections are equal in size.
8. Consider an  $7 \times 7$  grid shown in Figure 1 but ignore the different signs for grid points for now.
- (\*\*) How many paths (of length 12) are there from top left to lower right?
  - (\*) How many of them pass through the middle point, the one marked with  $\circ$ ?
  - (\*\*\*) How many paths from part (a) don't enter into the lower half of the grid, i.e. don't use any of points marked with  $\odot$ ?
9. If we flip a fair coin 4 times, what is the chances that it lands on the same side all the times?
10. If we flip a fair point 4 times, what is the expected (average) number of times it lands heads?
11. (\*) We flip a fair point 4 times and write down which side it landed. Call  $X$  the number of times it landed heads twice in a row. For example if the sequence was head, head, head, tail,  $X = 2$ . If it is head, tail, head, head,  $X = 1$ . What is the expected (average) of  $X$ ? What if we flipped the coin 100 times instead?

Figure 1: A  $7 \times 7$  grid.

12. We construct a graph on 4 vertices  $v_1, v_2, v_3, v_4$  by flipping a fair coin for each possible edge and adding it to the graph if the coin landed heads.
- (\*) What is the expected (average) number of simple paths from  $v_1$  to  $v_2$  of length 2? Remember that a path is called simple if it doesn't pass through any vertex more than once.
  - (\*) How about the expected number of simple paths from  $v_1$  to  $v_2$  (of any length)?
  - (\*\*) What would be the answer to the previous two questions if the graph had  $n$  vertices?
13. (\*) There are 20 businesspeople on the elevator in a 10 stories building. Each of them chooses one of the 9 floors  $2, \dots, 10$  to go completely at random and independently of others (from observation businesspeople go to random floors all the time.) What is the expected number of floors the elevator will not stop at?
14. Write a C, C++, Java or Python program that reads an integer from standard input (console) and prints the sum of all integers between 1 and  $n$  inclusive to standard output.  $n$  is small enough to fit in a 32-bit integer.

Make sure your program works for the following:

- $n = 1$ ; it should output 1.
- $n = 2$ ; it should output 3.
- $n = 1,000,000,000$ ; it should output 500,000,000,500,000,000.<sup>1</sup>
- $n = -5$ ; it should output  $-14$ .<sup>2</sup>

<sup>1</sup>i.e. the output might not fit in a 32 bit integer. Also, adding all the numbers with a for loop might be a bit too slow.

<sup>2</sup>i.e. the formula  $n(n+1)/2$  assumes that  $n > 0$ .