

## COMP 3804 — Assignment 2

**Due:** Thursday February 15, 23:59.

### Assignment Policy:

- Your assignment must be submitted as one single PDF file through Brightspace.

Use the following format to name your file:

LastName\_StudentId\_a2.pdf

- **Late assignments will not be accepted. I will not reply to emails of the type “my internet connection broke down at 23:57” or “my scanner stopped working at 23:58”, or “my dog ate my laptop charger”.**
- You are encouraged to collaborate on assignments, but at the level of discussion only. When writing your solutions, you must do so in your own words.
- Past experience has shown conclusively that those who do not put adequate effort into the assignments do not learn the material and have a probability near 1 of doing poorly on the exams.
- When writing your solutions, you must follow the guidelines below.
  - You must justify your answers.
  - The answers should be concise, clear and neat.
  - When presenting proofs, every step should be justified.

**Question 1:** Write your name and student number.

**Question 2:** Since you all miss COMP 2804 so much, let's start with a basic probability question. First some notation: If  $x_1, x_2, \dots, x_m$  are real numbers, then their product is written as

$$\prod_{i=1}^m x_i = x_1 \cdot x_2 \cdot x_3 \cdots x_m.$$

Let  $(S, \Pr)$  be a probability space, let  $m \geq 2$  be an integer, and let  $B_1, B_2, \dots, B_m$  be a sequence of events in this space. In this question, you will prove that

$$\Pr(B_1 \cap B_2 \cap \cdots \cap B_m) = \Pr(B_1) \cdot \prod_{i=2}^m \Pr(B_i \mid B_1 \cap B_2 \cap \cdots \cap B_{i-1}). \quad (1)$$

**Question 2.1:** Using the definition of conditional probability, prove that (1) holds when  $m = 2$ .

**Question 2.2:** Using the definition of conditional probability, prove that (1) holds when  $m = 3$ .

**Question 2.3:** Using induction and the definition of conditional probability, prove that (1) holds for every integer  $m \geq 2$ .

**Question 3:** In class, we have seen the following randomized selection algorithm:

**Algorithm** RSELECT( $S, k$ ):

**Input:** Sequence  $S$  of numbers, integer  $k$  with  $1 \leq |S| \leq k$

**Output:**  $k$ -th smallest number in  $S$

**if**  $|S| = 1$

**then** return the only element in  $S$

**else**  $p =$  uniformly random element in  $S$ ;

by scanning  $S$  and **making**  $|S| - 1$  **comparisons**, divide it into

$S_{<} = \{x \in S : x < p\}$ ,

$S_{=} = \{x \in S : x = p\}$ ,

$S_{>} = \{x \in S : x > p\}$ ;

**if**  $k \leq |S_{<}|$

**then** RSELECT( $S_{<}, k$ )

**else if**  $k \geq 1 + |S_{<}| + |S_{=}|$

**then** RSELECT( $S_{>}, k - |S_{<}| - |S_{=}|$ )

**else** return  $p$

**endif**

**endif**

**endif**

Let  $T$  be the random variable whose value is the **number of comparisons** made by this algorithm. With  $n$  denoting the length of the sequence  $S$ , we have shown that the expected value of  $T$  is  $O(n)$ .

A natural question to ask is if the value of  $T$  is  $O(n)$  with high probability, i.e., does there exist a positive constant  $C$ , such that

$$\lim_{n \rightarrow \infty} \Pr(T \leq Cn) = 1?$$

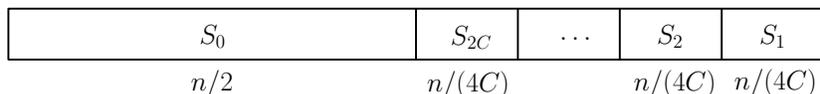
In this question, you will prove that the answer is “no”.

**In the rest of this question, we assume that the sequence  $S$  contains the numbers  $1, 2, 3, \dots, n$  in sorted order. We are going to run algorithm  $\text{RSELECT}(S, 1)$ .**

Algorithm  $\text{RSELECT}(S, 1)$  and its recursive calls choose pivots  $p_1, p_2, p_3, \dots$ :

- $p_1$  is chosen uniformly at random in  $\{1, 2, \dots, n\}$ .
- If  $p_1 \neq 1$ , then  $p_2$  is chosen uniformly at random in  $\{1, 2, \dots, p_1 - 1\}$ .
- If  $p_2 \neq 1$ , then  $p_3$  is chosen uniformly at random in  $\{1, 2, \dots, p_2 - 1\}$ .
- If  $p_3 \neq 1$ , then  $p_4$  is chosen uniformly at random in  $\{1, 2, \dots, p_3 - 1\}$ .
- Etcetera.

Let  $C$  be an arbitrary positive integer, and let  $n$  be a very large integer that is a multiple of  $4C$ . Divide the set  $\{1, 2, \dots, n\}$  into  $2C + 1$  subsets, as indicated in the figure below. The subset  $S_0$  has size  $n/2$ , whereas each subset  $S_i$ , for  $1 \leq i \leq 2C$ , has size  $n/(4C)$ .



Define the events

$$A = “T > Cn”$$

and, for each  $i = 1, 2, \dots, 2C$ ,

$$B_i = “p_i \in S_i”.$$

**Question 3.1:** Prove that

$$\Pr(B_1 \cap B_2 \cap \dots \cap B_{2C}) \leq \Pr(A).$$

**Question 3.2:** Use Question 2 to prove that

$$\Pr(A) \geq (1/(4C))^{2C}.$$

**Question 3.5:** Conclude that

$$\Pr(T \leq Cn) \leq 1 - (1/(4C))^{2C}.$$

**Question 4:** You are given a min-heap  $A[1 \dots n]$  and a variable *largest* that stores the largest number in this min-heap.

In class, we have seen algorithms  $\text{INSERT}(A, x)$  (which adds the number  $x$  to the min-heap and restores the heap property) and  $\text{EXTRACTMIN}(A)$  (which removes the smallest number from the heap and restores the heap property).

Explain, in a few sentences, how these two algorithms can be modified such that the value of *largest* is correctly maintained. The running times of the two modified algorithms must still be  $O(\log n)$ .

**Question 5:** Let  $m$  be a large integer and consider  $m$  non-empty sorted lists  $L_1, L_2, \dots, L_m$ . All numbers in these lists are integers. Let  $n$  be the total length of all these lists.

Describe an algorithm that computes, in  $O(n \log m)$  time, two integers  $a$  and  $b$ , with  $a \leq b$ , such that

- each list  $L_i$  contains at least one number from the set  $\{a, a + 1, \dots, b\}$  and
- the difference  $b - a$  is minimum.

For example, if  $m = 4$ ,

$$\begin{aligned} L_1 &= (2, 3, 4, 8, 10, 15) \\ L_2 &= (1, 5, 12) \\ L_3 &= (7, 8, 15, 16) \\ L_4 &= (3, 6), \end{aligned}$$

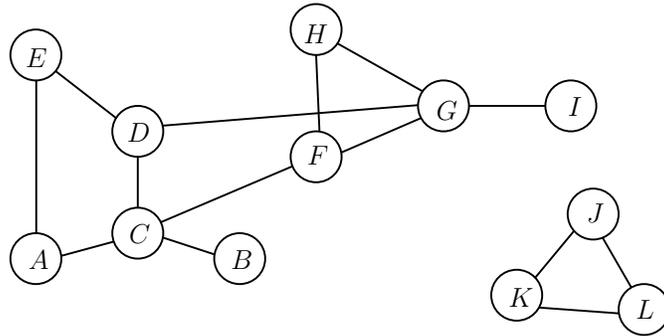
then the output can be  $(a, b) = (4, 7)$  or  $(a, b) = (5, 8)$ .

As always, justify the correctness of your algorithm and explain why the running time is  $O(n \log m)$ .

*Hint:* Use a min-heap of size  $m$  and use Question 4. For the example, draw it like this, then stare at it until you “see” the algorithm:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	2	3	4				8		10					15	
1				5							12				
						7	8							15	16
	3				6										

**Question 6:** Consider the following undirected graph:



**Question 6.1:** Draw the DFS-forest obtained by running algorithm DFS on this graph. Recall that algorithm DFS uses algorithm EXPLORE as a subroutine.

In the forest, draw each tree edge as a solid edge, and draw each back edge as a dotted edge.

Whenever there is a choice of vertices (see the two lines labeled (\*)), pick the one that is alphabetically first.

**Question 6.2:** Do the same, but now, whenever there is a choice of vertices (see the two lines labeled (\*)), pick the one that is alphabetically last.

```

Algorithm DFS( $G$ ):
  for each vertex  $u$ 
  do  $visited(u) = false$ 
  endfor;
   $cc = 0$ ;
  for each vertex  $v$  (*)
  do if  $visited(v) = false$ 
    then  $cc = cc + 1$ 
      EXPLORE( $v$ )
    endif
  endfor

```

```

Algorithm EXPLORE( $v$ ):
   $visited(v) = true$ ;
   $ccnumber(v) = cc$ ;
  for each edge  $\{v, u\}$  (*)
  do if  $visited(u) = false$ 
    then EXPLORE( $u$ )
  endif
  endfor

```

**Question 7:** Prove that an undirected graph  $G = (V, E)$  is bipartite if and only if  $G$  does not contain any cycle having an odd number of edges.

**Question 8:** Since Taylor Swift and Travis Kelce miss each other very much, they decide to meet. Taylor and Travis live in a connected, undirected, non-bipartite graph  $G = (V, E)$ . Taylor lives at vertex  $s$ , whereas Travis lives at vertex  $k$ .

Taylor and Travis move in steps. In each step, Taylor must move from her current vertex to a neighboring vertex, and Travis must move from his current vertex to a neighboring vertex.

Prove that there exists a moving strategy such that Taylor and Travis meet each other at the same vertex.

*Hint:* While moving around in the graph, each of Taylor and Travis may visit the same vertex more than once, and may traverse the same edge more than once.

If the graph  $G$  consists of the single edge  $\{s, k\}$ , then they will never be at the same vertex. But in this case  $G$  is bipartite.

If the graph contains a path having 8 edges, Taylor lives at one end-vertex, and Travis lives at the other end-vertex, will they ever be at the same vertex?

Question 7 is useful.