COMP 3804 — Assignment 2

Due: Monday February 28, 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:** Let $A[1 \ldots n]$ be an array storing $n$ numbers. We have seen algorithm BUILDHEAP($A$) that rearranges the numbers in the input array $A$ such that the resulting array is a max-heap; see page 56 of my handwritten notes. This algorithm uses the HEAPIFY-procedure as a subrouting; see page 53 of my handwritten notes. Consider the following variant of this algorithm:

```
Algorithm BUILDHEAP'(A):
    for $i = 1$ to $\lfloor n/2 \rfloor$
        do HEAPIFY($A, i$)
    endfor
```

Give an example of an array $A[1 \ldots n]$, where $n$ is a small integer (such as $n = 7$), which shows that algorithm BUILDHEAP' may not result in a max-heap.

**Question 3:** Let $A[1 \ldots n]$ be an array storing $n$ pairwise distinct numbers, and let $k$ be an integer with $0 \leq k < n$. We say that this array is $k$-sorted, if for each $i$ with $1 \leq i \leq n$, the entry $A[i]$ is at most $k$ positions away from its position in the sorted order.

For example, a sorted array is 0-sorted. As another example, the array

$$A[1 \ldots 10] = [1, 4, 5, 2, 3, 7, 8, 6, 10, 9]$$

is 2-sorted, because each entry $A[i]$ is at most 2 positions away from its position in the sorted order. For $i = 3$, $A[3]$ is 2 positions away from its position, 5, in the sorted array. For $i = 9$, $A[9]$ is 1 position away from its position, 10, in the sorted array.

Describe an algorithm SORT that has the following specification:

```
Algorithm SORT($A, k$):
    Input: An array $A[1 \ldots n]$ of $n$ pairwise distinct numbers and an integer $k$ with $2 \leq k < n$. This array is $k$-sorted.
    Output: An array $B[1 \ldots n]$ containing the same numbers as the input array. The array $B$ is sorted.
    Running time: Must be $O(n \log k)$.
```

Explain why your algorithm is correct and why the running time is $O(n \log k)$.

*Hint:* Use a min-heap of a certain size.
Question 4: You are given three beer barrels $B_1$, $B_2$, and $B_3$. Barrel $B_1$ has a capacity of 8 litres, barrel $B_2$ has a capacity of 5 litres, and barrel $B_3$ has a capacity of 3 litres.

At any moment, each barrel contains a given amount of beer (in litres). In one step, you can pour beer from one barrel, say $B_i$, to another barrel, say $B_j$. This step terminates at the moment when $B_i$ becomes empty or $B_j$ becomes full, whichever happens first.

To give some examples:

• If $B_1$ contains 6 litres of beer, $B_2$ contains 2 litres of beer, and $B_3$ contains 0 litres of beer, then we can pour the entire contents of barrel $B_2$ to barrel $B_3$. At the end of this step, $B_1$ contains 6 litres of beer, $B_2$ contains 0 litres of beer, and $B_3$ contains 2 litres of beer.

• If $B_1$ contains 3 litres of beer, $B_2$ contains 4 litres of beer, and $B_3$ contains 1 litre of beer, then we can pour 2 litres of beer from $B_1$ to $B_3$. At the end of this step, $B_1$ contains 1 litre of beer, $B_2$ contains 4 litres of beer, and $B_3$ contains 3 litres of beer.

Decision problem:

• Let $b_1$, $b_2$, and $b_3$ be integers such that $b_1 \geq 0$, $b_2 \geq 0$, $0 \leq b_3 \leq 3$, and $b_1 + b_2 + b_3 = 4$. Similarly, let $b'_1$, $b'_2$, and $b'_3$ be integers such that $b'_1 \geq 0$, $b'_2 \geq 0$, $0 \leq b'_3 \leq 3$, and $b'_1 + b'_2 + b'_3 = 4$.

• Initially, barrel $B_1$ is filled with $b_1$ litres of beer, barrel $B_2$ is filled with $b_2$ litres of beer, and barrel $B_3$ is filled with $b_3$ litres of beer.

• We want to decide whether or not it is possible to perform a sequence of steps that results in barrel $B_1$ having $b'_1$ litres of beer, barrel $B_2$ having $b'_2$ litres of beer, and barrel $B_3$ having $b'_3$ litres of beer?

(4.1) Formulate this as a problem on a directed graph. What are the vertices of the graph? What are the directed edges of the graph?

(4.2) Draw the entire graph.

(4.3) Assume that $(b_1, b_2, b_3) = (4, 0, 0)$ and $(b'_1, b'_2, b'_3) = (3, 1, 0)$. Use your graph to decide whether the answer to the decision problem is YES or NO.

(4.4) Assume that $(b_1, b_2, b_3) = (4, 0, 0)$ and $(b'_1, b'_2, b'_3) = (2, 1, 1)$. Use your graph to decide whether the answer to the decision problem is YES or NO.
**Question 5:** Let $G = (V, E)$ be an undirected graph. A *vertex coloring* of $G$ is a function $f : V \rightarrow \{1, 2, \ldots, k\}$ such that for every edge $\{u, v\}$ in $E$, $f(u) \neq f(v)$. In words, each vertex $u$ gets a “color” $f(u)$, from a set of $k$ “colors”, such that the two vertices of each edge have different colors.

Assume that the graph $G$ has exactly one cycle with an odd number of vertices. (The graph may contain cycles with an even number of vertices.)

What is the smallest integer $k$ such that a vertex coloring with $k$ colors exists? As always, justify your answer.

**Question 6:** Let $G = (V, E)$ be a directed graph, which is given to you in the adjacency list format. Thus, each vertex $u$ has a list that stores all vertices of the set

\[ \{v : (u, v) \in E\}. \]

The backwards graph $G_b$ is obtained from $G$ by replacing each edge $(u, v)$ in $G$ by the edge $(v, u)$. In words, in $G_b$, we follow the edges of $G$ backwards.

Describe an algorithm that computes, in $O(|V| + |E|)$ time, an adjacency list representation of $G_b$. As always, justify your answer and the running time of your algorithm.

**Question 7:** Let $G = (V, E)$ be a directed acyclic graph, and let $s$ and $t$ be two vertices of $V$.

Describe an algorithm that computes, in $O(|V| + |E|)$ time, the number of directed paths from $s$ to $t$ in $G$. As always, justify your answer and the running time of your algorithm.