Algorithms

Swarthmore Honors Exam, Spring 2022

Instructions and conventions.

• This exam is closed-book and closed-notes and closed-technology.

• If you are asked to design an algorithm, your answer should be clear and convincing. In particular, (a) you should either provide pseudocode or a very clear English description of the algorithm, and (b) you should explain rigorously why your algorithm is correct and has the desired/claimed running time, proving any non-obvious claims.

• You will be graded both on correctness and clarity. Please be concise and neat.

• Feel free to use as a black box any algorithm that you have seen in CS41.

• There are five problems on this exam, and a total of 50 points. Do as many of them as you can.

• Good luck!

Problems.

1. (10 points) Suppose you have \( n > 0 \) coins with distinct integer values \( c_1, c_2, \ldots, c_n \), so that \( c_i > 0 \) for all \( i \). Give an algorithm that takes as input the values \( c_1, \ldots, c_n \) as well as an integer \( k \geq 0 \), and outputs the number of ways to divide the coins into two piles so that both piles have value at least \( k \). Express the running time of your algorithm in terms of \( n \) and \( k \). The running time of your algorithm should be linear in \( n \). Get the best dependence on \( k \) that you can.

For example, if \( n = 3 \) and \( k = 2 \), and the three coins have values \( c_1 = 1, c_2 = 2, c_3 = 5 \), then your algorithm should output 4, since there are four ways to split up the coins into two piles so that each pile has total value at least 2:

<table>
<thead>
<tr>
<th>Pile 1</th>
<th>Pile 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,2}</td>
<td>{5}</td>
</tr>
<tr>
<td>{5}</td>
<td>{1,2}</td>
</tr>
<tr>
<td>{2}</td>
<td>{1,5}</td>
</tr>
<tr>
<td>{1,5}</td>
<td>{2}</td>
</tr>
</tbody>
</table>
2. (10 points) Let $G = (V, E)$ be an undirected connected graph with non-negative edge weights. Let $T \subseteq V$ be a set of terminals. A Steiner tree for $T$ is a subgraph of $G$ that is a tree and that contains all of the terminals in $T$, possibly along with some other vertices. The weight of a Steiner tree is the total weight on all of its edges.

(a) Describe an $O(n \log n + m)$-time algorithm that takes as input $G$ and a set $T$ of size two, and finds a minimum weight Steiner tree for $T$.

(b) Describe an $O(n^2 \log n + nm)$-time algorithm that takes as input $G$ and a set $T$ of size three, and finds a minimum weight Steiner tree for $T$.

(c) Describe an $O(n^2 \log n + nm)$-time algorithm that takes as input $G$ and a set $T$ of size four, and finds a minimum weight Steiner tree for $T$.

3. (10 points) You’ve just moved into a new dorm with a very strange culture.\(^\dagger\) Some of the folks living in the dorm are TruthTellers (T), while some are Unpredictable (U). You can’t tell which is which, but you can pair up two people and ask each of them about each other. A TruthTeller will always correctly report the type of their partner (T or U); an Unpredictable person will behave unpredictably, and may say either T or U no matter who they are paired with.

Suppose that there are $n$ people living in the dorm, and that strictly more than $n/2$ of them are TruthTellers. Give an algorithm that queries at most $O(n)$ pairs, and that will identify who is a TruthTeller and who is Unpredictable.

4. (10 points) There are $n$ honors exams at Swarthmore today. Say that exam $i$ is scheduled to begin at time $a_i$ and end at time $b_i$. Suppose that two exams that overlap in time cannot be administered in the same room. (Formally, we say that exam $i$ and exam $j$ are overlapping if $[a_i, b_i] \cap [a_j, b_j] \neq \emptyset$). Design an algorithm that takes as input arrays $A$ and $B$ of length $n$ so that $A[i] = a_i$ and $B[i] = b_i$ for $i = 1, \ldots, n$, and outputs the smallest number of rooms that are necessary to schedule all of the exams, as well as an optimal assignment of exams to rooms.

For example, if $A = [12\text{pm}, 4\text{pm}, 2\text{pm}]$ and $B = [3\text{pm}, 6\text{pm}, 5\text{pm}]$, then we need two rooms. Exam 1 and 2 can be scheduled in Room 1, while Exam 3 will be scheduled in Room 2.

Your algorithm should run in time $O(n \log n + nk)$, where $k$ is the minimum number of classrooms needed.

\(^\dagger\)Or possibly they are just playing an elaborate prank on you, but in either case you still have to do the problem.
5. **(10 points)** Suppose that there are $m$ computer science courses to be offered next semester at Swarthmore. Each course can have up to three ninjas. There are $n$ students interested in ninja-ing, and each student can ninja for at most one course. For each course $i$, there is a list $Q_i \subset \{1, \ldots, n\}$ of students that are qualified/interested in being a ninja for course $i$.

(a) Give an algorithm that, given the lists $Q_i$, assigns qualified ninjas to courses in a way that maximizes the total number of ninjas assigned. Your algorithm should run in time polynomial in $n$ and $m$.

(b) Suppose that $|Q_i| \geq 3$ for each $i$ (there are at least three qualified students for each course), and also that each student $j$ appears in at most two of the lists $Q_i$. Prove that your algorithm from part (a) will assign three ninjas to every course. (That is, the courses will all be fully staffed).