COMP 360 - Winter 2018 - Assignment 1

Due: Jan 27th, 11:59pm

In solving these questions you may consult books or other available notes, but you need to provide citations in that case. You can discuss high level ideas with each other, but each student must find and write his/her own solution. Copying solutions from any source, completely or partially, allowing others to copy your work, will not be tolerated, and will be reported to the disciplinary office.

You should upload the pdf file (either typed, or a clear and readable scan) of your solution to mycourses.

---

1. True or False? Explain your answer in one line.
   - (4 points)
   $$\forall S \subseteq \{1,2,\ldots,10\},\ \sum_{i\in S} i \le \sum_{i:2i\in S} i.$$
   - (4 points) $\log_2(2^5 n) = 5\log_2 n$.
   - (4 points) Let $G=(V,E)$ be an undirected graph. If $\forall u \in V\ \exists uv \in E$, then G is connected.
   - (4 points) Multiplying the capacities of all edges in a flow network by 2, multiplies the value of the maximum flow by 2.
   - (4 points) Consider a flow network with exactly one minimum cut (A,B). Suppose that there are k edges going from A to B. If we increase the capacities of every edge in the network by 1, then the value of the minimum cut will increase by k.
2. (15 points) Construct a flow network with n nodes that has 2^{n-2} minimum cuts. (This shows that no efficient algorithm can output all the minimum cuts, as simply there might be too many of them).
3. (15 Points) Consider the following two algorithms for finding the maximum flow:

Algorithm 1: Scaling max-flow

- Initially set $f(e) := 0$ for all edges e.
- Set Δ to be $\max c_e$ rounded down to a power of 2.
- While $\Delta \ge 1$:
- While there is an s,t-path P in $G_f(\Delta)$:
- Augment the flow using P and update $G_f(\Delta)$.
- Endwhile.
- Set $\Delta := \Delta/2$.
- Endwhile.
- Output f.

Algorithm 2: The fattest path algorithm

- Initially set $f(e) := 0$ for all edges e.
- While there exists an s,t-path in G_f:
- Augment the flow using the fattest s,t-path P in G_f.
- Update G_f.
- Endwhile.
- Output f.

In the second algorithm the fattest means the largest bottleneck. From the class we know that the number of augmentations in Algorithm 1 is at most $2m\lceil \log_2 K \rceil$, where K is the maximum capacity of an edge. Deduce from this that the number of augmentations in Algorithm 2 is also at most $2m\lceil \log_2 K \rceil$.

4. (a) (15 points) Show that for every flow network, there exists an execution of the Ford-Fulkerson algorithm that finds the maximum flow after at most m augmentations. Here m is the number of the edges of G.

   (b) (10 points) On the other hand, construct examples of flow networks that require $\Omega(m)$ augmentations no matter how we choose the augmenting paths. (If you are not familiar with the notation. see the definition of big Omega in Chapter 2.2 of the textbook).

5. Suppose that we have solved the Max Flow problem on a flow network $(G = (V, E), s, t, \{c_e\}_{e \in E})$, and found the flow $f : E \to \mathbb{R}$ with the largest value.

   (a) (10 Points) Someone increases the capacity of one of the edges by 1. Can we update the value of the Max-Flow in $O(m)$? (Note that to achieve this running time, we cannot afford to run FF from scratch).

   (b) (15 Points) Someone decreases the capacity of an edge that is adjacent to s by 1. Can we update the value of the Max-Flow in $O(m)$? (Note that to achieve this running time, we cannot afford to run FF from scratch).