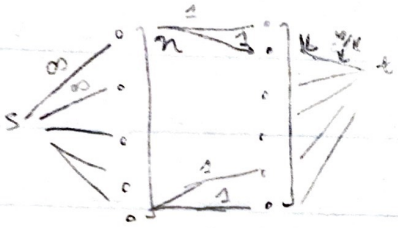
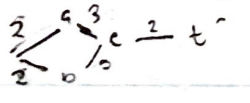
Jonathan Tso

CS 401 Homework 5

1. We can think of this similar to network flow, where we have n nodes with capacity 1 and k hospitals with capacity n/k



This is similar to the setup done in class, where we now have a flow from s and limit the flow from s to n with 1 (patient) and then from hospitals to t with capacity n/k. Because this is similar to the Bipartite matching problem discussed in class, I believe Ford-Fulkerson can also be used to solve this problem.

2. Consider best case scenario, where adding in 1 to each node n increases max flow. Then, our α is $(n)(1)$. The same applies to β, where $(n)(1)$: the reduced max flow. Now consider the following: 

When we all capacity + 1 to all, we only increase by 1, but if we decrease by 1, we have a loss of 2. Thus, $\alpha \leq \beta$

3. $A = \{a_1, \ldots, a_n\}$

$B_1, B_2 \ldots$ contains subsets of A

This problem is similar to vertex cover. Consider that given K, we look for the "at most" which means it is our constraint. This is similar to finding a minimum vertex cover in optimization of vertex cover. We span over all B's and look for interactions that align with every single value of H's subset, so that we have to go through each of H and look at all connecting B's to see if it fits the K constraint.

This NP complete because we can also have a verifier/certifier. If given an H, we can just check all the elements inside to see if it holds true.

4. n processes

m resources

- If process given all resources, it is active. Else, blocked
- Want to maximize number of active resources
- K minimum always active process?

- The resource problem is NP hard. Effectively, it is similar to the independent set problem, where each process is looking to see if there is independent set that will allow it to take all those resources for itself. We do this for all n processes. Since we know independent set is NP-complete, we can say this is thus NP-complete since we can reduce it.
- If $K=2$, we can perform this solution. We are effectively looking at any 2 processes over all m resources. If we see they have no resources to be shared, then we can say at least they can be concurrently run or not.
- This simplifies the original resource problem so that each n process only needs 2 resources (both are specific, though). However, we still are looking for independent sets in each of the simplified processes, just on a smaller scale. Thus, the problem should still be NP-complete since we can reduce it from independent set