COMP 7270 Assignment 4 200 points No late submissions!

Due by 11:59 PM Tuesday 04/25

Upload your submission well before this deadline. Even if you are a few minutes late, as a result of which Canvas marks your submission late, your assignment may not be accepted.

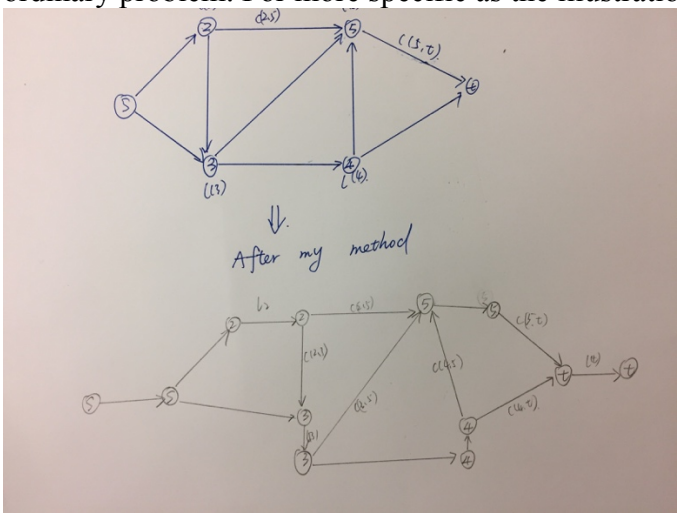
Instructions:

1. This is an individual assignment. You should do your own work. **Any evidence of copying will result in a zero grade and additional penalties/actions.**
2. Late submissions **will not** be accepted unless prior permission has been granted or there is a valid and verifiable excuse.
3. **Think carefully; formulate your answers, and then write them out concisely** using English, logic, mathematics and pseudocode (no programming language syntax).
4. Algorithms should be provided in numbered pseudocode steps.
5. **Type your answers in this Word document and submit it. If that is not possible, use a word processor to type your answers as much as possible (you may hand-write/draw equations and figures), turn it into a PDF document and upload**.

1. (26-1) in CLRS book (Escape Problem).

Answer:

a: First, we split each vertex into two vertices, connect them with a new edge. Then, we use the previous vertex capacity as the edge capacity. However, since the network flow is a directed graph, we should be careful about the order of vertices. Suppose the new two vertices V_1 and V_2 , divided by V previously , has direction as V_1 to V_2. So the last vertex which connect V previously now should be connect to V_1. So the new edge could be used in a same function as the vertex capacity, which maximum flow come into the part should be not bigger than this capacity.so it can be has same size like before. Because the source and sink vertex is the beginning and ending part, so it is meaningless if they has capacity. Like, if sink has capacity, it should be the maximum number because other vertex can not beyond. So there is verity that maximum flow network problem with vertex and edge capacity could be reduced into an ordinary problem. For more specific as the illustration below:



Note: $\mathcal{L}$(number) is the vertex capacity, c(u,v) is the edge capacity.

B: we could use maximum network flow with edge and vertices capacity.
First, we create an source vertex and connect it with those starting vertices.
Second, we create a sink vertex and connect it with 4n-4 boundary point.
Third, we consider all grid edge as directed edges for which all edges has same positive capacity, then we find edge-disjoint path, if the size of the maximum flow in the resulting graph is equal or bigger than m then there exist m edge-disjoint path , which means it has the escape path. otherwise it do not have escape path.
According to the running time, first we need to reduce our process into the ordinary maximum flow problem. Like the first question we split each vertices into two which need O(V), so the number of new vertex should be two times of the old one, also we need to add more |V| edges , if we use Ford-Fulkerson method to find the maximum flow , suppose the time is "f", so the total time for running this algorithm is $O[f^*(|E|+|V|)]=O(n^3)$

2. (26-2) in CLRS book (Minimum path cover)
Answer:
A: we use maximum flow algorithm to find the maximum bipartite matching selects the edges to use in a minimum path cover. Define a graph G' with source x_o and y_o ,also define all edges capacity to be 1.G' is the flow network corresponding to a bipartite graph G' with two parts: L and R with $L=(x_1,x_2,\ldots x_n)$ and $R = (y_1,y_2\ldots y_n)$
So we can construct a path cover P from Bipartite matching M by moving from some x_i to the matching y_i so as follows:
(1) start a new path containing a vertex i that has not yet been placed in a path.
(2)if x_i is unmatched, the path can't go any further: just add it to P.
(3) if x_i is matched to some y_i add j to the current path . If j has already been placed in a path, combine the path with that one and go back to step2. Otherwise go to step 2 to process x_j

B: this algorithm is not suit for cycles one . There is counterexample that a G with u vertices, a triangle with three directed edge, and also has a extra vertex point outside to one of the vertex. E = {(1,2),(2,3),(3,1),(4,1)}, G could be covered in a single path : 4-1-2-3, but our algorithm might find only a 2–path cover

3. (26-3) in CLRS book (Algorithmic consulting).
Answer:
a : Suppose we have some $J_i \in T$ and $A_k \notin T$ for every $A_k \in R_i$. Meanwhile, if $A_k \notin T$ so

$A_k \in S$, so the cut(S,T) should on the edge between A and J. However, according the definition of question, $c(A_k,J_i)=\infty$ if $A_k \in R_i$, which means the given(S,T)of G would have a infinite capacity, which is contradict to the given cut is finite-capacity cut. So our assumption is contracted here. Hence the cut(S,T)will be finite capacity cut if $A_k \in T$ for every $A_k \in R_i$.

b: If we want to get the maximum net revenue ,we need this following equation that : (total income of accepted jobs – cost to employing the experts). Suppose the set of accepted jobs is (P.J), and (P.A) is the set of experts hired in. so we will get the maximum when all the hired experts are required for the jobs. The edge from s to A_k is representing the cost $c(s,A_k)=c_k$ and the edge from vertices j_i to sink t would represent income,
that is $c(J_i, t)=p_i$. so our desired revenue "r" is $r = \sum_{Ji \in J} Pi$ -c(S,T). since the c(S,T) = $\sum_{(s,Ak) \in EA} c(s, Ak)=\sum_{(s,Ak) \in JA} ck$
similarly for the edges of the form(J_i,t) the possibility that$(J_i,t) \in E_j$ is true only when $J_i \in S$ which means $J_i \notin T$, which in turn proves that $J_i \notin P.J$, so $\sum_{(j,t) \in Ej} c(Ji, t)=\sum_{(J,t) \in P.J} Pi$
so the revenue is:
$r = \sum_{(J,t) \in P.J} Pi - \sum_{(s,Ak) \in JA} Ck$
$= \sum_{Ji \in J} Pi$ - c(S,T)

In short, the maximum revenue problem can be transported into the minimum cut. The maximum revenue for the project model is: $\sum_{Ji \in J}$ pi - c(S, T).

c: we could create a flow graph according the source s , sink t , and $(s,A_i),(A_n,J_m)$ and (J_m,t).,all the edge have capacity c_i,$c(A_k,J_i)$,and p_i, respectively. So to find the right job selection and expert for the job first is to find maximum flow ,when we use some algorithm to find the maximum, we can also get the minimum cut . If any expert belong to the minimum cut set but not hired yet , then hire the expert .If a job is not in the minimum cut then get it .

Algorithm-selection-maximum-revenue()
1.use Ford-Fulkerson algorithm to find the maximum flow and residual network
2.Find the minimum cut set(S,T)
3.if $E(s,A_k) \in (S,T)$
4.hire the expert
5.if $E(J_i,t) \notin (S,T)$
6.Accept the job
7.end

4. (26-4) in CLRS book (Updating maximum flow).
Answer:
A: if we execute the maximum-flow algorithm we could find edge(u,v)in E with increased capacity ensures that edge(u,v) is in the residual graph. So we can find augmenting path. There are two cases we need to consider: if (u,c) is or is not an edge that across a minimum cut. Because we do not change the value of the maximum flow. If (u,v) cross a minimum cut ,then increasing its capacity by 1 increase the capacity of that minimum cut by 1. So the maximum flow should increased by 1. So we just use one iteration of Ford-Fulkerson suffice. Run in O(E+V)

B: if f is the maximum flow before reducing C(u,v), if f(u,v) = 0, we do not need to change
If f(u,v)>=1, we will need to update the maximum flow.
Find a path from s to t which contains (u, v) in O(V + E). Decrease every edge of the flow by 1,which will decrease total flow by 1, then run an iteration of the Ford-Fulkerson algorithm to look for an augmenting path in graph in O(V+E), to see if we can find the augmenting path.

5. (26.4-3) in CLRS book.
Answer: For each vertex$\in E_f$, the number of relabel operation is no more than 2|V|-1. Also, edge (u,v) will be examined during relabel operations at most u|V|-2=O(V)times ,at most 2|V|-1 for calling to relabel(u) and (v) respectively, so there are at most O(E), so the total time would be O(VE)

Now, you may be tired of flows. Try one more questions for brain exercise.

6. (26.4-6) in CLRS book.

Answer: relabel operation and saturating pushes has the same times to run . The most pushing time for an edge to becomes saturated is k, so the number of non-saturating push time is $2k|V||E|$, thus the total number of basic operation is at most $2|V|^2 + 2|V||E|+2k|V||E|= O(k\ VE)$