CS2223
HW#5

DUE: Tuesday, November 28

1. (6 points) Suppose you want to find a *cheapest* path (path of minimal cost) in a weighted graph $G=(V,E)$, $w:E\to\Re^+$, from a vertex $\sigma\in V$ to a vertex $\tau\in V$, which is short and does not contain many edges. In particular, the *cost* of a path consisting of edges $e_1,...,e_k$ is $k+\sum_{1\le i\le k} w(e_i)$, and a *cheapest* path is a path of minimal cost. Describe an algorithm to find a cheapest path in worst case time in $O\left(|V|^2\right)$.

2. (5 points) Consider the following algorithm which operates on a weighted graph $G=(V,E)$, $w:E\to\Re^+$. We assume that the graph is *complete* and that all edge weights are distinct. That is, there is an edge between every pair of distinct vertices and for each $e_0,e_1\in E, e_0\ne e_1$, $w(e_0)\ne w(e_1)$.

```
MAKE-EMPTY-(MAX)-PRIORITY-QUEUE(Q)
for each v ∈ V − {σ}
        INSERT(v,Q) with priority D if σv∈E then w(σv) else -∞
D[σ]←0
while Q!=∅
        v ← EXTRACT-MAX(Q)
        for each u∈Q do
                D[u] ← max{D[u], D[v] + w(v,u)}
```

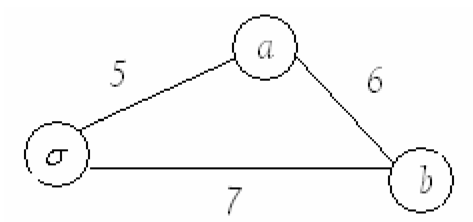
Does this algorithm always compute the length of a longest simple path from the source, σ, to every other vertex in *V*? (No vertex is traversed more than once by a simple path.) Justify your response.

3. (8 points) Suppose that a weighted graph $G=(V,E)$, $w:E\to\Re$ is connected, but note that the weights may be negative. Describe an algorithm to construct a minimum spanning subgraph (it must be connected, but it may admit cycles).

CS2223
HW#5 SOLUTIONS

1. A **cheapest** path in G is a **shortest** path in $H=(V,E)$ with weight function $w'(e)=w(e)+1$ for all $e \in E$. So we use Dijkstra's Algorithm to solve this problem in H with any of the data structures we discussed, since they all work in time in $O\left(|V|^2\right)$.

2. The algorithm does not always compute the length of longest simple path from the source, σ, to every other vertex in V. For the graph



the algorithm would return $D[b]$=7, whereas the length of a longest simple path from σ to b has length 11.

3. Every $e \in E$ with $w(e)<0$ belongs to a minimum spanning subgraph. So all we do is add to the edges of negative weight enough edges of positive weight to span V (connect the subgraph).

$T \leftarrow \{e \in E \mid w(e)<0\}$
for each $uv \in E, w(uv)<0$ **do**
MERGE(u,v)
while T is not connected **do**
(v,w) := DELETEMIN(Q);
if FIND(v)<>FIND(w) **then begin**
MERGE(v,w)
$T \leftarrow T \cup \{(v,w)\}$
end
end

EXTRACT (& remove from E) the shortest edge $e \in E$

if $T \cup \{e\}$ is acyclic **then** $T \leftarrow T \cup \{e\}$