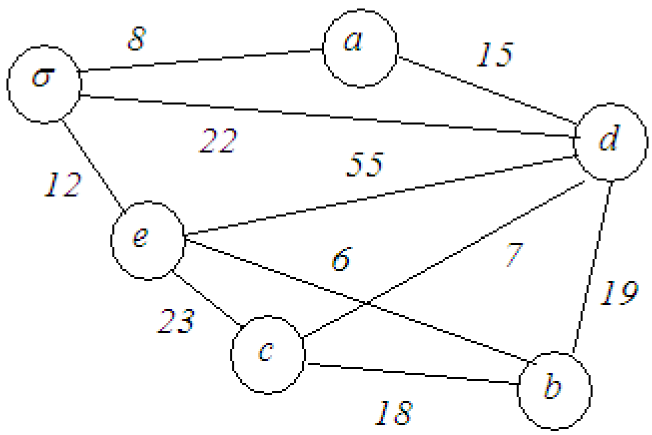
CS584
HW#4

DUE: Wednesday, November 11

1. (15 points) For any connected weighted graph $G=(V,E)$ with source $\sigma \in V$ and any path $P=(v_0,...,v_k)$ of G, the *BottleNeckWeight* of P is $\max_{0\le i<k}(w(v_i v_{i+1}))$, the maximum of the weights of all the edges on the path. For $v \in V$, the *BottleNeckDistance* of v is the minimum, over all paths from σ to v, of the *BottleNeckWeight* of the paths. So, in the graph below, the path (σ,a,d,b,c) has *BottleNeckWeight* 19, and path (σ,a,d,c) provides c with a *BottleNeckDistance* of 15.



v	*BottleNeckDistance*
σ	0
a	8
b	12
c	15
d	15
e	12

a Describe an efficient algorithm to find, for every $v \in V$, the *BottleNeckDistance* of v.
b Analyze your algorithm (with appropriate data structures).
c Prove that you algorithm works.

2. (12 points) In the class notes we saw that the shortest path problem can be solved by finding maximal values of d to satisfy BELLMAN'S EQUATIONS:

$$d[\sigma]=0$$

$$\forall u,v \in V \;\; d[v] \le d[u]+w(uv)$$

The problem is that it is difficult to find a good bound on the number of applications of the triangular inequality above. A systematic application of these equations becomes efficient if we add restrictions to G. Assume that G is a directed acyclic graph with INDEGREE(σ)=0. Give an algorithm which uses, in the worst-case,

$\Theta(n+m)=\Theta(|V|+|E|)$ applications of BELLMAN'S EQUATIONS find the length of a shortest path from σ to every vertex of *G*.

3. (5 points) Show how to reduce the 0/1-Knapsack problem to the problem of finding a longest path in a directed acyclic graph. That is, show
   · how to translate an instance of 0/1-Knapsack problem to an instance of the problem of finding a longest path in a directed acyclic graph, and then,
   · how to translate a longest path in the directed acyclic graph back to a solution of the corresponding Knapsack problem.

4. (2 points) Prove or give a counterexample to the following.
CONJECTURE: If the weight of every edge of a connected weighted graph is distinct, then the union of the shortest paths from σ to every other vertex comprise an MST of the graph.

5. (2 points) Prove or give a counterexample to the following.
CONJECTURE: In order to use Dijkstra's Algorithm to find the shortest simple paths from σ to every other vertex even if edges of the graph can have negative weights, we can add the same large number to every weight, find the shortest paths in the new graph, and then subtract the large number from all the edge weights.

The motivation behind this CONJECTURE is to to make sure that the modified graph does not contain any negative weight cycles.

6. (4 points) For any connected weighted graph $G=(V,E)$ and any edge $uv \in E$, give a worst-case $O(n^2)$ algorithm to find a shortest cycle containing uv.

CS584
HW#4
SOLUTIONS

1. ***a*** The *BottleNeckDistance* from σ to any $v \in V$ is the *BottleneckWeight* of the unique path from σ to *v* in an MST of *G*.

Use PRIM'S ALGORITHM to compute the MST of *G*

Do a BREADTHFIRSTSEARCH of the MST from σ, and when

$v_k v_j$ is traversed with v_k in the tree, $d[j] \leftarrow \min(d[k], w(v_k v_j))$

b Using a Fibonacci Heap in the implementation of PRIM'S ALGORITHM, we can implement it in $O(m + n \lg n)$ time. The BREADTHFIRSTSEARCH can be done in time in $O(n)$.

c When PRIM'S ALGORITHM applies the Red Rule to remove edge *uv*, *u* and *v* are connected by a path (not including *uv*) on which every edge has a weight $\leq w(uv)$. So *uv* can be removed without influencing the *BottleNeckDistance* of either *u* or *v*.

2. TOPOLOGICALSORT(*G*) $O(E)$ operations

(Assume that *V* and *E* are numbered so that $\sigma = v_1$ and $(i < j) \rightarrow (v_j v_i \notin E)$)

for $j \leftarrow 2$ **to** *n* **do**

$d[j] \leftarrow \infty$

for each $v_k v_j \in E$ **do**

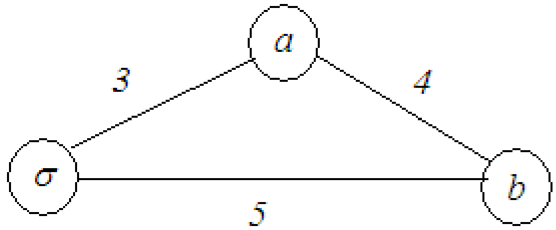
$d[j] \leftarrow \min(d[j], d[k] + w(v_k v_j))$

return *d*

The algorithm uses $\Theta(n+m) = \Theta(|V| + |E|)$ time.

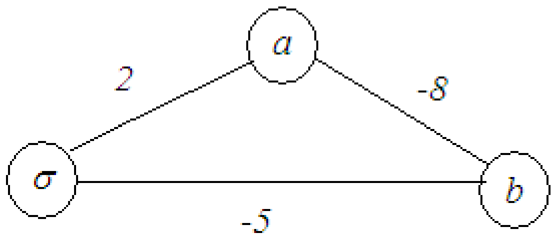
3. Given: *n*>0 objects, *v*[1..*n*], *w*[1..*n*], *W*, we construct an acyclic digraph with $n(W+1)+1$ vertices $\{u_j^x, 1 \leq j \leq n, 0 \leq x \leq W\} \cup \{\sigma\}$. Vertex u_j^x corresponds to a packing using only some subset of the first *j* objects, and into a Knapsack of capacity *x*. There is an edge from each u_j^x to u_{j+1}^x of weight 0 and, if $x + v[j+1] \leq W$, then there is an edge from u_j^x to $u_{j+1}^{x-w[j+1]}$ of weight $v[j+1]$. Finally, there is an edge from σ to u_1^W of weight 0 and, if $w[1] \leq W$, then there is an edge from u_j^x to $u_1^{x-w[1]}$ of weight $v[1]$. An optimal packing is the length of a longest/heaviest path from σ to u_n^0.

4. The union of the edges of the shortest paths from σ to *a* and *b* of graph



is the set $\{\sigma a, \sigma b\}$, but the MST consists of the edges $\{\sigma a, ab\}$

5. The CONJECTURE is false. The shortest simple path from σ to *b* is (σ, a, b).



But after adding the same large number to every weight, the shortest simple path becomes (σ, b).

6. The shortest cycle is edge *uv* plus the shortest path from source *u* to *v* in $G = (V, E - \{uv\})$.