Problem Set No. 5 *Corrected*

1. (Heaps with addition) **Heaps with addition** support the following operations:

   make-heap(h)**:** Create a new, empty heap named h.

   insert(i, k, h)**:** Insert item i, with real-valued key k, into h.

   delete-min(h)**:** Delete and return an item of minimum key from heap h.

   meld(h_1, h_2)**:** Combine item-disjoint heaps h_1 and h_2 into a single heap, named h_1; h_2 becomes empty.

   add-to-keys(k, h)**:** Add real number k to the key of *every* item in heap h.

   Describe an implementation of heaps with addition such that the amortized time for each operation is $O(1)$, except for **delete-min**, which has an amortized time bound of $O(\log n)$ on a n-item heap.

2. (Directed Minimum Spanning Trees)

   Consider a directed graph G with real-valued edge costs and with a distinguished root vertex r such that every vertex is reachable from r. A *minimum spanning tree* of G is a directed tree rooted at r, containing all the vertices, of minimum total edge cost. Consider the following algorithm for finding such a tree:

   1. (Find cheap edges) For each vertex other than r, choose a minimum-cost incoming edge. If the selected edges define an acyclic subgraph, return this subgraph as the MST. Otherwise, go to Step 2.
   2. (Shrink a cycle of cheap edges) Let $C = v_0, v_1, v_2, \ldots, v_k, v_{k+1} = v_0$ be a cycle of selected edges. Form a new graph by shrinking $v_0, \ldots, v_k$ into a single new vertex v, replacing each edge of the form (u, v_i) of cost $c(u, v_i)$ by a new edge (u, v) of cost $c'(u, v) = c(u, v_i) - c(v_{i-1}, v_i)$, and deleting loops. (The shrunken graph may contain multiple edges; we allow these.)
   3. Find an MST T' in the shrunken graph G' by applying the algorithm recursively. Expand T' to an MST in the original graph G by replacing the unique edge of the form (u, v) in T by the corresponding edge (u, v_i) in G and adding every edge on cycle C *except* (v_{i-1}, v_i).

   (a) Describe how to implement this algorithm to run in $O(n \log n + m)$ time on an n-vertex, m-edge graph. Hint: Use the solution to Problem 1.

   (b) (Extra Credit) Prove that this algorithm correctly finds a minimum spanning tree.

3. (Heuristic Search) Let G be a directed graph with real-valued edge costs and with a start vertex s and target vertex t. We wish to find a shortest path from s to t. Consider the following version of the labeling and scanning single-source shortest path algorithm discussed in class, where $e(v)$ is an easily-computed function that estimates the shortest distance from v to t:

   Initialization: set $d(s) = 0, d(v) = \infty$ for $v \neq s,\ L = \{s\}$.

   Repeat the following steps until L is empty (t is not reachable from s) or t is removed from L:

   > Delete from L a vertex v such that $d(v) + e(v)$ is minimum. For each edge (v, w) such that $d(v) + c(v, w) < d(w)$, replace $d(w)$ by $d(v) + c(v, w)$, replace $p(w)$ by v, and add w to L if it is not already present.

   The hope is that when t is removed from $L, d(t)$ will be the length of a shortest path from s to t, and this path may be recovered by following p-pointers backward from t. We wish to find conditions on the estimate e that guarantee that this is the case. Suppose that $e(v) - c(v, w) \leq e(w)$ for every edge (v, w), and that $e(t) = 0$. Then we say that e is a *safe* estimate.

   (a) Prove that if e is a safe estimate then, for any vertex v, $e(v)$ is a lower bound on the length of a shortest path from v to t. Prove that if there is a negative cycle from which t is reachable, then *no* estimate is safe.

   (b) Prove that if e is a safe estimate, then the shortest path algorithm removes every vertex from L at most once, and when a vertex v is removed from $L, d(v)$ is the actual shortest distance from s to v.

   (c) Suppose e and f are both safe estimates, with $e(v) \geq f(v)$ for every vertex v. Prove that, when the algorithm is run with estimate e, no more vertices are removed from L than when the algorithm is run with estimate f.

   (d) Consider an infinite augmented grid graph whose vertices are the lattice points (i, j) with i and j integral, having bidirectional edges $\{(i, j), (i, j + 1)\}, \{(i, j), (i + 1, j)\}, \{(i, j),$

$(i+1, j+1)\}$, and $\{(i,j),(i+1,j-1)\}$ for all integers i,j. The cost of the horizontal and vertical edges is exactly 1, and the cost of each diagonal edge is at least 1. Assume that there is a function $c((i,j),(k,l))$ that in constant time returns the cost of the edge $(\{(i,j),(k,l)\}$. Let the start vertex $s = (a,b)$, and let the target vertex $t = (0,0)$. consider the two distance estimates $f(x,y) = 0$ and $e(x,y) = \max\{|x|,|y|\}$. Prove that these are both safe estimates. As a function of $|x|$ and $|y|$, to within a constant factor, give the running time of the shortest path algorithm is it uses estimate e. Repeat for the estimate f.