Homework 7

Robbie McKinstry, Jack McQuown, Cyrus Ramavarapu

19 September 2016

Problem 8:

Given an arbitrary tree $\mathcal{T}$ with both positive and negative edge weights, finding the maximally weighted simple path, $\mathcal{P}$ can be found recursively by considering both the best linear path and the best kinked path available at a given vertex v.

Although this seems to suggest that naively recursing to the immediate subproblem, in this case a sub tree $\mathcal{T}'$ of $\mathcal{T}$, and extracting the two required values is sufficient, such thinking actually proves to be problematic. This is because although a path may not be the immediate best choice, its inclusion into the solution may prove beneficial at a later stage because it leads to an overall greater edge weight.

Accounting for the possibility that a path may not be immediately optimal, gives rise to the following recursive algorithm that considers both the linear and kinked paths at a given vertex.

```
**Function:** $MWSP$
**Input:** $Node\ \mathcal{N}$
**if** $\mathcal{N}$ $has\ no\ children$ **then**
    $return\ 0$
**else**
    $linear\ 1,\ linear\ 2 = max\_2(\forall\ children\ i:\ \ MWSP(i) + w_{parent,i})$
    $best\ linear = max(linear\ 1, linear\ 2)$
    $L.append(best\ linear)$
    $kinked\ path = linear\ 1 + linear\ 2$
    $K.append(kinked\ path)$
    **Return:** $L,\ K$
**end**
```

The maximum element from $L,\ K$ can be taken. If this maximum value is less than 0, the maximum weighted simple path can be taken to be 0 since not not moving from any vertex will be optimal.

Although the recursive algorithm produces the maximum weight of a simple path on a tree, an interative process can be developed that will run in linear time by starting with the leaves and moving up the tree while keeping track of the kinked and linear paths at every vertex.

```
**Function:** $IT\ MWSP$
**Input:** $Tree\ \mathcal{T}$
**Globals:** $A\ [\ ][\ ]$
$j = 0$
**foreach** $Leaf\ \ Node\ v$ **do**
    $A\ [v][0] = 0$
    $A\ [v][1] = 0$
    $j++$
**end**
/* Assume the nodes are numbered from the leaves upward  */
**for** $j\ \ to\ n$ **do**
    $linear\ 1,\ linear\ 2 = max\_2(\forall\ children\ i : A[j][i] + w_{j,i})$
    $A\ [j][0] = max(linear\ 1,\ linear\ 2)$
    $A\ [j][1] = linear\ 1 + linear\ 2$
**end**
```

As with the recursive algorithm, the maximum value between the two rows of A will be the answer, unless this value is negative. In this case the answer will be 0 for the reason mentioned above.

Problem 9: