1. Show how the version of the dynamic trees data structure in Figure 3 of JST19 changes if you do the following operations: \texttt{addcost(f,-2)}, \texttt{[foo, fooCost]=findcost(x)}, \texttt{cut(foo)}, \texttt{link(foo,c)}, \texttt{addcost(foo,fooCost)}. 

Please print out this form (two-sided, if you can) and write your answers legibly in the spaces provided. If you can’t write legibly, type.
2. Consider the residual graph shown below. Draw an instance of the dynamic trees data structure that might be used by Dinic’s algorithm to represent subtrees with non-zero residual capacity. Make vertices $c$, $e$ and $t$ tree roots. Include as many edges in the trees as you can. Show the costs of all non-root vertices.

![Residual Graph Diagram]

3. Suppose you implemented the dynamic trees data structure using nothing but an array of parent pointers. What is the worst-case running time of each of the operations, for this implementation?