Solutions for CS174 Homework 8

- The probability that a min cut of the original graph G with n vertices survives as a min cut of the contracted graph G' with t vertices is $(t/n)^2$. Let $P(t)$ be the probability that a call to the algorithm with t vertices successfully computes the min cut (given that the min cut is still in the graph). So similar to the lecture notes, $P(t) = 1 - (1 - \frac{1}{4}P(\frac{t}{2}))^2 = \frac{1}{2}P(\frac{t}{2}) - \frac{1}{16}P(\frac{t}{2})^2$. The recurrence has two terms on the RHS, but as $P(m)$ decreases which it does as n increases, the second term is negibly small. The solution to the recurrence $P(t) = 1/2P(t/2)$ is easily shown by induction to be $P(t) = \Theta(\frac{1}{n})$.

- For a graph with $O(n)$ edges, the *total work* to contract all the edges is $O(n)$ using a suitable data structure (e.g. union-find). So from the divide-and-conqure algorithm, we have the formula $T(n) = 2T(n/\sqrt{2}) + O(n)$. We know the following lemma holds:

  Lemma 0.1. *Suppose $T(n)$ satisfies the recurrence relation: $T(n) = aT(n/b) + \Theta(n^\alpha)$, if $n > 1$, $T(n) = \Theta(1)$, if $n = 1$. Let $\beta = \log_b a$. Then $T(n) = \Theta(n^\beta)$, if $\alpha < \beta$.*

  It's easy to see that the lemma holds by drawing a recursion tree. So $T(n) = O(n^2)$.

- Note that all the edges that are T-light are actually in T. Because if an edge $e = (u, v)$ is T-light and not in T, then we can always remove the edge with the highest weight in the unique path from u to v in T and add e. The resulting tree will be a spanning tree with less weight than T. This contradicts the assumption that T is a minimum spanning tree. T contains $n - 1$ edges. Therefore the number of edges in G that are T-heavy is $m - n + 1$.