HW: Core Proofs

Copyright Oliver Serang, 2018
University of Montana Department of Computer Science

1. You are sorting an array of \( n \) objects. The objects only support the <
operator and the == operator. Prove that any algorithm that sorts this
array is \( \Omega(n \log(n)) \).

There are \( n! \) possible arrangements of the array. There are \( \log(n!) \) comparisons needed to sort it.

\[
\log(n!) = \log(n) + \log(n-1) + \cdots + \log(1) > \log(n) + \log(n-1) + \cdots + \log(\frac{n}{2}) + 0 + 0 + \cdots + 0
\]

\[= \frac{n}{2} \log(\frac{n}{2}) = \frac{n}{2} \left( \log(n) - \log(2) \right) = \Omega(n \log(n))
\]

2. You are constructing a suffix tree on a string of length \( n \). Assume that
you have access to a fast fastscan routine. Prove that the runtime of
all slowscan calls will be \( \in O(n) \).

Given \( |head_{p-1}| = k, |head_p| > k-1 \) via suffix lemma.

Thus fastscan descends \( k-1 \) characters and slowscan
descends \( |head_p| - (|head_{p-1}| - 1) \) in that iteration.

Thus, over every iteration the cost is

\[\in \sum_{i} |head_p| - |head_{p-1}| + 1\]

\[= |head_n| - |head_{n-1}| + 1 + |head_{n-1}| - |head_{n-2}| + 1 \cdots\]

\[= |head_n| - |head_0| + n \]

\( \in O(n) \)
3. You are solving TSP on a metric graph. Prove that you can achieve a 2-approximation using the MST.

\[ G = \text{graph} \]
\[ p^* = \arg \min_{p \in \text{paths}(G)} \left( \text{cost}(p) \right) \]
\[ T^* = \arg \min_{T \in \text{tree}(G)} \left( \text{Cost}(T) \right) \]

\[ \text{Cost}(T^*) \leq \text{cost}(p^*) \]  
// paths harder than trees

\[ \text{Cost}(p^*) \leq 2 \cdot \text{cost}(T^*) \]  
// via rewiring edges

\[ \text{Cost}(p^{(m)}) \leq \text{cost}(p^*) \]  
// via reducing \( b_1 \geq b_2 \) type edges with \( b_1 \geq b_2 \) by metric property

\[ \text{Cost}(p^{(m)}) \leq \text{cost}(p^*) \leq 2 \cdot \text{Cost}(T^*) \leq 2 \cdot \text{Cost}(p^*) \]