CSE 6140 (Fall 2017)

Computational Science and Engineering Algorithms

Prof. A. Benoit
Georgia Institute of Technology

LaTeXer: W. Kong
http://wwkong.github.io
Last Revision: September 13, 2017

Table of Contents

These notes are currently a work in progress, and as such may be incomplete or contain errors.

ACKNOWLEDGMENTS:

Special thanks to *Michael Baker* and his LaTeX formatted notes. They were the inspiration for the structure of these notes.

Abstract

The purpose of these notes is to provide the reader with a secondary reference to the material covered in CSE 6140.

Administrative

1 Algorithms

1.1 Greedy Algorithm

[Lecture 1+2]

Definition 1.1. A greedy algorithm is an algorithm in which we always progress by making locally optimal choices.

[Lecture 3, Greedy pt. 2, slide 8]

Theorem 1.1. *[Interval scheduling] Suppose that we are given a greedy solution $A = \{a_1, ..., a_k\}$ and optimal solution $O = \{o_1, ..., o_m\}$ for $m \geq k$. Assume that $f_{o_i} < f_{o_j}$ and $f_{a_i} < f_{a_j}$ for $i < j$. The greedy algorithm for interval scheduling is optimal, i.e. $f_{a_r} \leq f_{o_r}$ for $r \leq k$.*

Proof. (by induction) Base case is trivial. Suppose that $f_{a_{r-1}} \leq f_{o_{r-1}}$ Then

$$\begin{aligned} & f_{a_{r-1}} \leq f_{o_{r-1}} \leq s_{o_r} \leq f_{o_r} \\ \implies & o_r \text{ is compatible with } a_1, ..., a_{r-1} \\ \implies & o_r \text{ was an option for greedy} \\ \implies & a_r \text{ was chosen by greedy} \\ \implies & f_{a_r} \leq f_{o_r} \end{aligned}$$

and so we show that $m = k$. Assume that $k < m$. Our claim states that $f_{a_k} \leq f_{o_k}$. The optimal solution must have a o_{k+1} and since it is feasible,

$$\begin{aligned} & f_{o_r} \leq s_{o_{k+1}} \\ \implies & o_{k+1} \text{ is a an option for greedy after iteration } k \\ \implies & \text{contradicts with greedy stopping at iteration } k \\ \implies & k = m \end{aligned}$$

□

[Lecture 4, Greedy pt. 3]

Theorem 1.2. *Dijkstra's algorithm is correct.*

Proof. For each $u \in S$, we are given the invariant $d(u)$ which is the length of the shortest $s \mapsto u$ path. We proceed by a proof by induction. The base case is $|S| = 1, S = \{s\}$, and $d(s) = 0$ so we are optimal.

Suppose that when $|S|$, the inversion holds and for all $u \in S$, $d(u)$ is the shortest path length. Consider the next node v added to S by Dijkstra, i.e. $S \hookleftarrow S \cup \{v\}$.

Now v must have edge (u, v) such that $u \in S$ and

$$\pi(v) = d(u) + e_{uv}$$

and $\pi(v)$ is the best choose, by greediness. We now must show that $d(v) = \pi(v) = d(u) + e_{uv}$ is the shortest path length. Consider any $s \mapsto v$ path p, then p must leave S and there is an edge (x, y) with $x \in S, y \notin S$. Now

$$\begin{aligned}
\pi(p) &\geq \pi(p') + e_{xy} \\
&\geq d(x) + e_{xy} && \text{(since } x \in S) \\
&\geq \pi(y) && \text{(by definition)} \\
&\geq \pi(v) && \text{(by greedy choice)}
\end{aligned}$$

for any path, $\pi(p) \geq \pi(v) = d(v)$ and $d(v)$ is shortest. □

[Lecture 5, MCST]

Cut Property (proof)

Theorem 1.3. *We are given a graph (E, V) with distinct edge weights. Let S be any set of nodes and let $e = (u, v)$ be the minimum cost edge from the cut set of S. Then every MST / MCST T^* contains e.*

Proof. (exchange argument) Given e, T^*, and S, suppose that e does **not** belong to T^*. Without loss of generality (WLOG) suppose that $u \in S$ and $v \in V \backslash S$. Then there exists a path in T^* which connects u and v, since T^* is a spanning tree, which is not e, and it must cross from S to $V \backslash S$ at some edge $f = (a, b)$.

Exchange f for e to get $T' = T^* - f + e$. Clearly T' has $n - 1$ edges, so let us argue it is connected. When we remove f and add e, there is now a new path $a \to u \to v \to b$ instead of $u \to a \to b \to v$. Hence, T' is a MCST with smaller cost which is impossible. Hence $e \in T^*$ for any MCST T^*. □

(Lecture 6, Recursion)

Theorem 1.4. *Given*

$$T(n) \leq \begin{cases} 0, & n = 1 \\ T\left(\lceil \frac{n}{2} \rceil\right) + T\left(\lfloor \frac{n}{2} \rfloor\right) + n, & n > 1. \end{cases}$$

We have $T(n) \leq n \lceil \log n \rceil$.

Proof. By induction. The base case is trivial. Suppose that it is true for $k \leq n-1$ and consider $k = n$. Let us define $n_1 = \lfloor n/2 \rfloor$ and $n_2 = \lceil n/2 \rceil$. Then,

$$\begin{aligned}
T(n) &\leq T(n_1) + T(n_2) + n \\
&\leq n_1 \lceil \log(n_1) \rceil + n_2 \lceil \log(n_2) \rceil + n \\
&\leq (n_1 + n_2) \lceil \log(n_2) \rceil + n \\
&= n \lceil \log(n_2) \rceil + n.
\end{aligned}$$

What remains is to prove that $\lceil \log(n_2) \rceil \leq \lceil \log(n) \rceil - n$ (simple math). □
