Notes of the Introduction To Algorithms

Kai Zhao

November 23, 2016

Contents

Part I

Foundations

Chapter 1

The Role of Algorithms in Computing

1.1 Algorithms

Exercies

1.1-1 Give a real-world example that requires sorting or a real-world example that requires computing a convex hull.

Answer: One example that requires sorting is that teachers will sort our scores after the exam.

1.1-2 Other than speed, what other measures of efficiency might one use in a real-world setting ?

Answer: cost, space, manpower, material resources. In different cases, each can be the key of meausres of efficiency.

Reference: https://www.quora.com/Other-than-speed-what-other-measures-of-efficiency-might-one-use-in-a-real-world-setting

1.1-3 Select a data structure that you have seen previously, and discuss its strengths and limitations.

Answer: Array

strengths: access directly

limitations: costs lot when insert or delete

1.1-4 How are the shortest-path and traveling-salesman problems given similar? How they are different?

Answer:

1.1-5 Come up with a real-world problem in which only the best solution will do. Then come up with one in which a solution that is "approximately" the best is good enough.

Answer:

1.2 Algorithms as a technology

Chapter 2

Getting Started

Chapter 3

Growth of Functions

Chapter 4

Divide-and-conquer

Chapter 5

Probabilistic Analysis and Randomized Algorithms

Part II

Sorting and Order Statistics

Chapter 6

Heapsort

Part III

Data Structures

Part IV

Advanced Design and Analysis Techniques

Part V

Advanced Data Structures

Part VI

Graph Algorithms

Chapter 7

Minimum Spanning Tree

7.1 Notes

(i) There maybe more than one MST in a forest.

(ii) The number of all the edges in the MST is equal to $\boldsymbol{V} - 1$.

7.2 Growing a minimum spanning tree

7.2.1 Definition

A

A is a subset of some minimum spanning tree.

Safe edge

Safe edge is a edge that add to A and A is also a subset of some minimum spanning tree.

7.2.2 Generic-MST

GENERIC-MST(G, w)
1 $A = \varnothing$
2 **while** A does not form a spanning tree
3 find an edge (u, v) that is **safe edge** for A
4 $A = A \cup \{(u, v)\}$
5 return A

Initialization: After line 1, the set A trivially satisfies the loop invariant.
Maintenance: The loop in lines 2-4 maintains the invariant by adding only safe edges.
Termination: All edges added to A are in a minimum spanning tree, and so the set A returned in line 5 must be a minimum spanning tree.

7.2.3 Theorem 1.

Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function ω defined on E. Let A be a subset of E that is inclued in some minimum spanning tree for G, let $(S, V - S)$ be any cut of G that respects A, and let (u, v) be a light edge crossing $(S, V - S)$. Then, edge (u, v) is **safe** for A. Namely, $A \cup (u, v)$ is also included in some minimum spanning tree for G.

Proof Let T be a minimum spanning tree that includes A, and assume that T does not contain the light edge (u, v), since if it does, the edge is obviously **safe** for A. We shall construct another minimum spanning tree $T^{'}$ that includes $A \cup (u, v)$ by using cut-and-paste technique, thereby showing that (u, v) is a **safe** edge for A.

The edge (u, v) forms a cycle with the edges on the simple path p from u to v in T. Since u and v are on opposite sides of the cut $(S, V - S)$, at least one edge in T lies on the simple path p and also crosses the cut. Let (x, y) be any such edge. The edge (x, y) is not in A, because the cut respects A. Since (x, y) is on the unique simple path from u to v in T, removing (x, y) breaks T into two components. Adding (u, v) reconnects them to form a new spanning tree $T^{'} = T - \{(x, y)\} \cup \{(u, v)\}$.

We next show that $T^{'}$ is a minimum spanning tree. Since (u, v) is a light edge crossing $(S, V - S)$ and (x, y) also crosses this cut, $w(u, v) \leq w(x, y)$. Therefore, $w(T^{'}) = w(T) - w(x, y) + w(u, v) \leq w(T)$.

When $w(T^{'}) == w(T)$, we know that $T^{'}$ is also a minimum spanning tree, so the edge (u, v) is **safe** for A.

When $w(T^{'}) < w(T)$, since we let T be a minimum spanning tree and **assume** that T does not contain the light edge (u, v). Therefore, the **assume** is false, so T must contain the light edge (u, v), and the edge (u, v) is **safe** for A.

7.2.4 Exercises

23.1-1

Let (u, v) be a minimum-weight edge in a connected graph G. Show that (u, v) belongs to some minimum spanning tree of G.

Solution

Let E_u be all the edges that connected to the point u.

a. If there is only one edge connected to the point u, the edge belongs to **all** the minimum spanning tree of G.

b. If there is more than one edge connected to the point u, we assume that (u, v) is not in any minimum spanning trees of G. There must be one edge $(u, x) x! = v$ that is in some minimum spanning tree of G, since $w(u, v) < w(u, x)$, therefore, the edge (u, x) can not be in some minimum spanning tree of G. So there is conflict and the assume is false. So, the (u, v) belongs to some minimum spanning tree of G.

23.1-2

Professor Sabatier conjectures the following converse of Theorem 1. in Minimum Spanning Tree. Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let $(S, V - S)$ be any cut of G that respects A,and let (u, v) be a safe edge for A crossing $(S, V - S)$. Then, (u, v) is a light edge for the cut. Show that the professorfis conjecture is incorrect by giving a counterexample.

Solution

a. Here is a special case, the point v of (u, v) only has one edge, and $w(u, v)$ is the largest, let (x, y) be any other edge that crosses the cut, obviously, (u, v) is not a light edge for the cut.

b. Here is a generic case, assume that there is a light edge $(u^{'}, v^{'})$ crossing the cut and the edge has no common point with (u, v), so $w(u^{'}, v^{'}) < w(u, v)$. After combine A with $(u^{'}, v^{'})$, there is another $cut^{'}$ that crossing (u, v), and it is a light edge for $cut^{'}$. The previous case shows that (u, v) is not a light edge for any cut but some cut when (u, v) is a safe edge for A.

23.1-3

Show that if an edge (u, v) is contained in some minimum spanning tree, then it is a light edge crossing some cut of the graph.

Solution

Let T be the minimum spanning tree that contains the edge (u, v), if we remove the edge from T, and the other edges are A, obviously there is some cut that crosses the edge (u, v) which respects A. Then we are going to show that the edge (u, v) is a light edge crossing these cut.

If there is only one edge crossing the cut, obviously the edge (u, v) is a light edge crossing the cut.

If there is more than one edge crossing the cut, let (x, y) be any edges crossing the cut other than (u, v). Assume that $w(x, y) < w(u, v)$, there will another minimum spanning tree $T^{'}$ and $w(T^{'}) = w(T) - \{(u, v)\} + \{(x, y)\} < w(T)$ which is impossible since the T is a minimum spanning tree. So the assume is contradiction and $w(x, y) \geq w(u, v)$, so the edge (u, v) is a light edge crossing some cut of the graph.

23.1-4

Give a simple example of a connected graph such that the set of edges $\{(u, v)$: there exists a cut $(S, V - S)$ such that (u, v) is a light edge crossing $(S, V - S)\}$ does not form a minimum spanning tree.

Solution

There is a quadrangle: $V = A, B, C, D, E = (A, B), (A, C), (B, C), (B, D), (C, D)$ $, w(A, B) = w(A, C) = w(B, C) = 1, w(B, D) = w(C, D) = 2$. Obviously, $(A, B), (A, C)and(B, C)$ are lights edges crossing some cut. So the tree edges can join the set. And they construct a circle, so the set can not form a minimum spanning tree.

I think if we add respect to the set, then the set will form a minimum spanning tee. Such as, $\{(u, v)$: there exists a cut $(S, V - S)$ which repects this set such that (u, v) is a light edge crossing $(S, V - S)\}$, and the set will form a minimum spanning tree.

23.1-5

Let e be a maximum-weight edge on some cycle of connected graph $G = (V, E)$. Prove that there is a minimum spanning tree of $G^{'} = (V, E - e)$ that is also a minimum spanning tree of G. That is, there is a minimum spanning tree of G that does not include e.

Solution

Assume that there is a minimum spanning tree T of G including the edge e. Firstly, we construct a tree $T^{'}$ same as T, and remove the edge e from $T^{'}$. There is another edge $e^{'}$ on the same cycle of G with e, and $T^{'}$ does not have a cycle after add $e^{'}$ to $T^{'}$. Since $w(e) \geq w(e^{'})$, so $w(T) \geq w(T^{'})$. Therefore, there is a minimum spanning tree $T^{'}$ that does not include e.

23.1-6

Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge crossing the cut. Show that the converse is not true by giving a counterexample.

Solution

1. Proof

Let T be the minimum spanning tree that is constructed by the unique light edges crossing each cut. Assume that $T^{'}$ is another minimum spanning tree which is different from T. We are going to show that the assume is contradiction that $T^{'}$ can not be a minimum spanning tree.

Let x be the vertex which has different edges in T and $T^{'}$. Let edge (x, y_1) be the edge in T but not in $T^{'}$. Let edge (x, y_2) be the edge in $T^{'}$ but not in T.

Now we are going to show that $w(x, y_1) < w(x, y_2)$**.** If we add the edge (x, y_2) into the T, there will be an cycle including the edge (x, y_1) and (x, y_2). Since there is a unique light edge for each cut, so $w(x, y_1)! = w(x, y_2)$. Assume $w(x, y_1) > w(x, y_2)$, so we will get a better minimum spanning tree after replace (x, y_1) with (x, y_2). Since T is a minimum spanning tree, so there can not be a better minimum spanning tree. So the assume that $w(x, y_1) > w(x, y_2)$ is contradiction. So $w(x, y_1) < w(x, y_2)$.

If we add the edge (x, y_1) into the $T^{'}$, there will an cycle including the edge (x, y_1) and (x, y_2). Since $w(x, y_1) < w(x, y_2)$, we can get a better minimum spanning tree if we replace (x, y_2) with (x, y_1). So the $T^{'}$ is not a minimum spanning tree. Therefore, the assume that $T^{'}$ is another minimum spanning tree which is different from T is contradiction.

2. Counterexample

$G = (V, E)$ has three vertex: A, B, C and two edges $(A, B), (A, C)$ which $w(A, B) =$

$w(A, C)$. There is a unique minimim spanning tree. However, the cut of $\{A\}, \{B, C\}$ does not have a unique light spanning tree.

23.1-7

Argue that if all edge weights of a graph are positive, then any subset of edges that connects all verticles and has minimum total weight must be a tree. Give an example to show that the same conclusion does not follow if we allow some weights to be nonpositive.

Solution

Firstly, we prove that the subset is a graph. Secondly, we prove that the subset does not contain a cycle.

1) Since the subset of edges connect all verticles, the subset must be a graph.

2) Assume there is a cycle in the subset, since all the weights are positive, if we remove one edge in the cycle, we will get a lesser total weight. However, the subset has minimum total weight, so the assume is contradiction. So, there is no cycle in the subset.

So the subset must be a tree.

Counterexample

$G = (V, E), V = \{A, B, C, D\}, E = \{(A, B), (B, C), (C, A), (A, D), (B, D), (C, D)\}$,
$w(A, B) = w(B, C), w(C, A) = -1, w(A, D) = 1, w(B, D) = 2, w(C, D) = 3$,
the minimum total weight is $w(A, B) + w(B, C) + w(C, A) + w(A, D) = -2$
but it has a cycle.

Corollary

All edge weights are positive, then any that connects all vertices and has minimum total weight must be a minimum spanning tree.

23.1-8

Let T be a minimum spanning tree of a graph G, and let L be the sorted list of the edge weights of T. Show that for any other minimum spanning tree $T^{'}$ of G, the list L is also the sorted list of edge weights of $T^{'}$.

Solution-1

We are going to replace different edges in $T^{'}$ with the same weight edges in T. If finally $T^{'}$ is the same as T, then we are done.

1 **while** find a vertex u in $T^{'}$ which only in two different edges in $T^{'}$ and T
2 Let (u, x) in $T^{'}$ but not in T
2 Let (u, y) in T but not in $T^{'}$
2 There is a cut $(S, V - S)$ which S includes x, y and excludes u. Since both the T and $T^{'}$ are minimum spanning tree, so $w(u, x) == w(u, y)$, so we can replace (u, x) with (u, y).
3 The final $T^{'}$ is the same as T, since each edge replaced in $T^{'}$ has the same weight as in T, so we are done.

Solution-2

Reference

Let list $A = a_1, a_2, ..., a_(i - 1), a_i, a(i + 1), ..., a_n$ be the sorted weights of T in ascending order.

Let list $B = b_1, b_2, ..., b_(i - 1), b_i, b(i + 1), ..., b_n$ be the sorted weights of T in ascending order.

Assume there is a difference weight between A and B which is the ith, so $a_i! = b_i$. We are going to show that the assume is contradiction. We are going to prove when the $a_i > b_i$, and it is also applied to $a_i < b_i$.

(1) If b_i in the list A, then there is a_j in the list A and $a_j == b_i$. Since $a_x == b_x$ when x is from 1 to $(i - 1)$, $j \geq i$, so $b_i == a_j \geq a_i$, so $b_i \geq a_i$, so the assume is contradiction.

(2) If b_i does not in the list A, there will a cycle in T when we add the edge of b_i. And b_i is not less than any other weights in the cycle. There is must a edge in the cycle which does not exist in $T^{'}$. Let the edge be a_x and $b_i \geq a_x$. Since $b_i \geq a_x \geq a_i$, the assume is contradiction.

Therefore, the assume is contradiction, so there is not a difference weight between A and B.

23.1-9

Let T be a minimum spanning tree of a graph $G = (V, E)$, and let $V^{'}$ be a subset of V. Let $T^{'}$ be the subgraph of T, and let $G^{'}$ be the subgraph of G induced by $V^{'}$. Show that if $T^{'}$ is connected, then $T^{'}$ is a minimum spanning tree of $G^{'}$.

Solution

Firstly, we will prove that **when $T^{'}$ is connected, if we replace $T^{'}$ with another tree $T_1^{'}$ that connects all the $V^{'}$ in $G^{'}$, the T of G will be T_1, and T_1 is also a tree**. Assume that there is cycle in T_1 after replace $T^{'}$ with $T_1^{'}$. Let $A, B, A^{'}, B^{'}$ in the cycle, and A, B is in V not in $V^{'}$, $A^{'}, B^{'}$ is in $V^{'}$, and $A^{'}$ is the first vertex that A connects $T^{'}$, $B^{'}$ is the first vertex that B connects $T^{'}$. Since A, B is in the cycle, there must be several edges that connect A and B. Also, there must be several edges connect $A^{'}$ and $B^{'}$. For all the trees in $G^{'}$ which connects all the $V^{'}$, $A^{'}$ connects $B^{'}$, so does the $T^{'}$, so there is also a cycle in T which is a minimum spanning tree. So the assume is contradiction.

Secondly, assume that there is a lesser weight tree than $T^{'}$ in $G^{'}$, then replace $T^{'}$ with it, we will get a lesser weight tree in G. Obviously, the assume is contradiction. So there is not a lesser weight tree in $G^{'}$ than $T^{'}$. So $T^{'}$ is a minimum spanning tree of $G^{'}$.

23.1-10

Given a graph G and a minimum spanning tree T, suppose that we decrease the weight of one of the edges in T. Show that T is still a minimum spanning tree for G. More formally, let T be a minimum spanning tree for G with edge weights given byy weight function w. Choose one edge $(x, y) \in T$ and a positive number k, and define the weight function $w^{'}$ by

$$w^{'}(u, v) = \begin{cases} w(u, v) & \text{if } (u, v) \neq (x, y), \\ w(x, y) - k & \text{if } (u, v) == (x, y). \end{cases}$$

Show that T is a minimum spanning tree for G with edge weights given by $w^{'}$.

Solution

Let (u, v) be the edge whose weight is decreased. If we remove the (u, v) in T, we can get two subtrees of T. Since only the weight of (u, v) is changed, the subtrees of T is also minimum spanning trees of its sub graph which is proved in **23.1-9**. So we should find a safe edge to connect to two subtrees which was (u, v). Obviously, the old weight of (u, v) is less or equal than any edges that connects the two subtrees. Since the new weight is less than the old weight, the (u, v) is also the safe edge for the two subtrees. So T is also a minimum spanning tree.

23.1-11

Give a graph G and a minimum spanning tree T, suppose that we decrease the weight of one of the edges not in T. Give an algorithm for finding the minimum spanning tree in the modified graph.

Solution

Add the decreased edge to the T, then there must be a cycle and remove the edge whose weight is the largest. DFS traverse the T whose time complexity is O(V).

TODO

TODO: Prim -¿ Kruskal

TODO: Kruskal -¿ Prim

7.3 The algorithms of Kruskal and Prim

Both **Kruskal** and **Prim** use a specific rule to **determine a safe edge** in line 3 of GENERIC-MST.

In Kruskal's algorithm, the set **A** is a forest whose vertices are all those of the given graph. The safe edge added to **A** is always a **least-weight** edge in the graph that **connects two distinct components**.

In Prim's algorithm, the set **A** forms a single tree. The safe edge added to **A** is always a **least-weight edge connecting the tree to a vertex not in the tree**.

7.3.1 Kruskal

7.3.2 Prim

Part VII

Selected Topics

Part VIII

Appendix: Mathematical Background