Introduction to Algorithms

Meng-Tsung Tsai

10/31/2019

Announcements

Midterm will be held in class on Nov 05 from 10:10 - 12:30.

Scope: slides 01 - 12, assignments, and their generalizations.

Programming Assignment 2 is due by Nov 10, 23:59. at https://oj.nctu.me

Matroid

What is a matroid?

It is a structure $M = (S, F)$ so that

(1) S is a set of n elements,
(2) F is a non-empty collection of subsets of S,

--- Example ---

Let $S = \{1, 2, ..., n\}$.

Then, F may be $\{\{1, 2\}, \{n\}, \emptyset, ...\}$.

What is a matroid?

It is a structure $M = (S, F)$ so that

(1) S is a set of n elements,
(2) F is a non-empty collection of subsets of S,
(3) F is *hereditary*; that is, if $A \subseteq B$ and $B \in F$, then $A \in F$.

--- Example ---

If $F = \{\{1, 2\}, ...\}$, then F also contains $\{1\}$, $\{2\}$, and $\varnothing$.

What is a matroid?

It is a structure $M = (S, F)$ so that

(1) S is a set of n elements,
(2) F is a non-empty collection of subsets of S,
(3) F is *hereditary*; that is, if $A \subseteq B$ and $B \in F$, then $A \in F$.
(4) M satisfies *exchange property*; that is, for every $A, B \in F$, if $|A| < |B|$, then there exists some $e \in B \setminus A$ so that $A \cup \{e\} \in F$.

--- Example ---

If $F = \{\{1, 2\}, \{n\}, ...\}$, then F also contains $\{1, n\}$ and $\{2, n\}$.

Weighted Matroid Problem

Weight Function

Let $\omega(e)$ be a function that assign each element e in S to a non-negative weight.

For any $R \subseteq S$, let $\omega(R) = \sum_{e \in R} \omega(e)$.

We say M is a *weighted matroid* if its S is associated with a non-negative weight.

Weighted Matroid Problem

Input: a matroid M and a weight function $\omega: S \rightarrow R_{+} \cup \{0\}$.

Output: a set R in F whose $\omega(R)$ is maximum among all sets in F.

--- Note ---

Many problems can be encoded as the weighted matroid problem, and all of them can be solved in a unified way.

A Greedy Algorithm for the Weighted Matroid Problem

```
Greedy(M = (S, F), ω){

    Sort elements in S by their weights into the non-increasing order;
    // Assume that s1, s2, ..., sn is the sorted order.

    A = ∅;
    for(i = 1; i ≤ n; ++i){
        if(A ∪ {si} in F){
            A ← A ∪ {si};
        }
    }
    return A;
}
```

--- Note ---

This algorithm can solve the weighted matroid problem optimally.

Applications

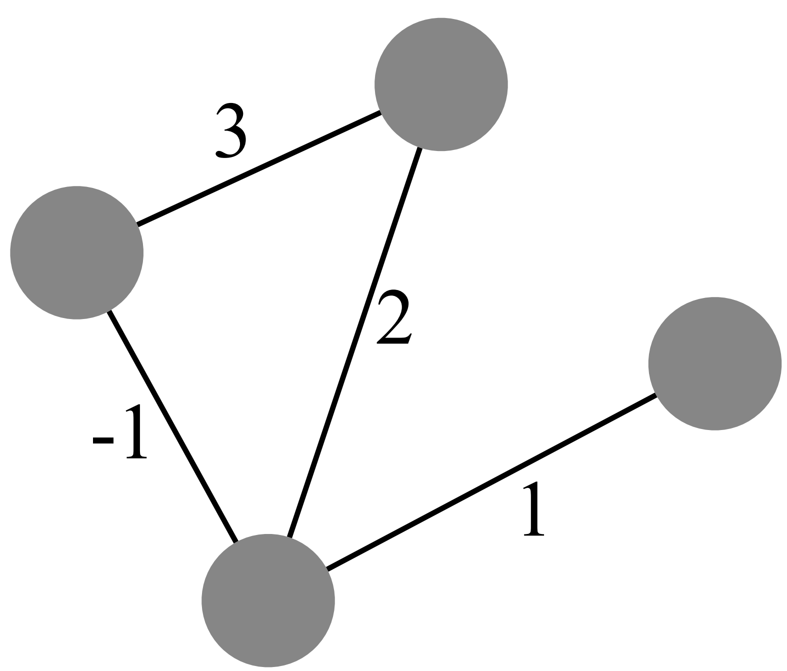
Minimum Spanning Trees

Input: an undirected graph G = (V, E) and a weight function $\omega: E \rightarrow R$.

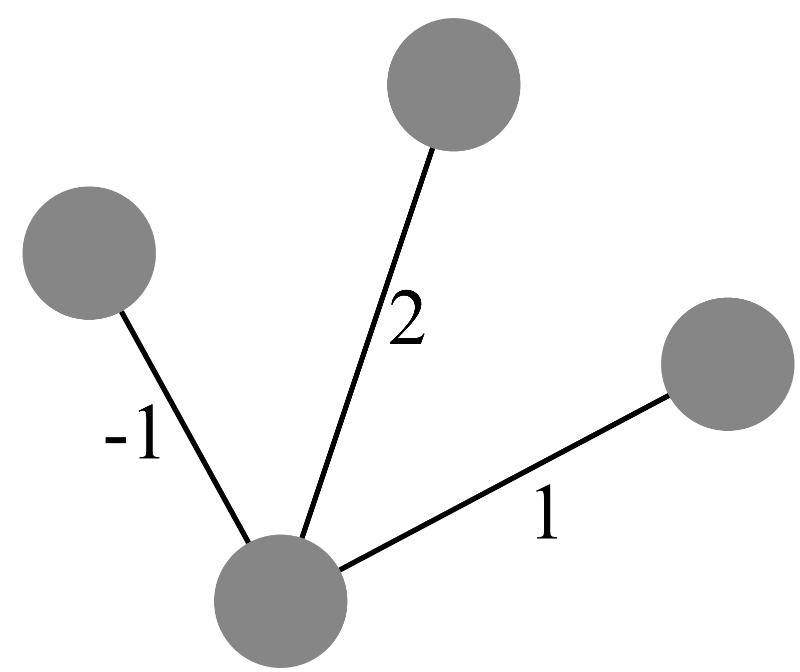
Output: a tree that spans all nodes in V so that the sum of the weight of edges in the tree is minimum among all such trees.

--- Example ---

graph G



MST



MST is weighted matroid problem

Plan: encode MST as a weighted matroid problem so that we can solve MST by the unifed greedy algorithm for WMP.

We define MST matroid as follows:

Let $M_G = (E, F)$ so that
(1) E is the set of all edges in G.
(2) F is the set of all spanning forests of G.
(3) Is F hereditary?

--- Answer ---

Suppose B is in F, then B has no cycle. A can be obtained from B by removing some edges, so A has no cycle. Thus A in F.

MST is weighted matroid problem

Plan: encode MST as a weighted matroid problem so that we can solve MST by the unifed greedy algorithm for WMP.

We define MST matroid as follows:

Let $M_G = (E, F)$ so that
(1) E is the set of all edges in G.
(2) F is the set of all spanning forests of G.
(3) Is F hereditary?
(4) Does M_G satisfy the exhange property?
--- Answer ---
Let A, B be two spanning forests in F so that $|A| < |B|$. Thus, # connected component in A is more than that in B. Hence, some CC in B contains two nodes from different CCs in A. There is an edge (x, y) in the path that connected u and v in B so that they belong to different CCs in A. Thus, $A \cup \{(x, y)\} \in F$.

MST is a weighted matroid problem

Plan: encode MST as a weighted matroid problem so that we can solve MST by the unifed greedy algorithm for WMP.

We define MST matroid as follows:

Let $M_G = (E, F)$ so that
(1) E is the set of all edges in G.
(2) F is the set of all spanning forests of G.
(3) Is F hereditary?
(4) Does M_G satisfy the exhange property?

In MST, we need to minimize the total weight. However, in WMP the total weight in maximized. We get this fixed by setting that

$$\omega'(e) = \omega_0 - \omega(e) \text{ where } \omega_0 = \max_{e \in S} \omega(e).$$

Task Scheduling Problem

Input: a set of unit-time task $\{s_1, s_2, ..., s_n\}$. Each task s_i has a deadline d_i and a non-negative penalty ω_i.

Output: a permutation $p_1p_2...p_n$ of elements in S so that p_i starts at time slot i-1 and finishes at time slot i and that the total penalty incurred by the tasks not finished by their deadlines is minimized.

--- Example ---

i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
ω_i	70	60	50	40	30	20	10

Task Scheduling Problem

Input: a set of unit-time task $\{s_1, s_2, ..., s_n\}$. Each task s_i has a deadline d_i and a non-negative penalty ω_i.

Output: a permutation $p_1p_2...p_n$ of elements in S so that p_i starts at time slot i-1 and finishes at time slot i and that the total penalty incurred by the tasks not finished by their deadlines is minimized.

--- Example ---

penalty = 50

i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
ω_i	70	60	50	40	30	20	10

TSP is a weighted matroid problem

Plan: encode TSP as a weighted matroid problem so that we can solve MST by the unifed greedy algorithm for WMP.

We define TSP matroid as follows:

Let $M_S = (S, F)$ so that
(1) S is the set of all tasks.
(2) F contains all subsets R of S so that all tasks in R can be done by their deadlines by an optimal arrangement.
(3) Is F hereditary?
--- Answer ---

If $A \subseteq B$ and all tasks in B can be solved by their deadlines, of course all tasks in A can.

TSP is a weighted matroid problem

Plan: encode TSP as a weighted matroid problem so that we can solve MST by the unifed greedy algorithm for WMP.

We define TSP matroid as follows:

Let $M_S = (S, F)$ so that
(1) S is the set of all tasks.
(2) F contains all subsets R of S so that all tasks in R can be done by their deadlines by an optimal arrangement.
(3) Is F hereditary?
(4) Does M_G satisfy the exhange property?
--- Answer ---

Yes, see Page 445 in I2A.

TSP is a weighted matroid problem

Plan: encode TSP as a weighted matroid problem so that we can solve MST by the unifed greedy algorithm for WMP.

We define TSP matroid as follows:

Let $M_S = (S, F)$ so that
(1) S is the set of all tasks.
(2) F contains all subsets R of S so that all tasks in R can be done by their deadlines by an optimal arrangement.
(3) Is F hereditary?
(4) Does M_G satisfy the exhange property?

The weighted matroid problem can find a subset R with maximum sum of penalty, so the penalty incurred by S \ R is minimized.

Why the greedy algorithm return the optimal solution?

Recall the greedy algorithm

```
Greedy(M = (S, F), ω){

    Sort elements in S by their weights into the non-increasing order;
    // Assume that s1, s2, ..., sn is the sorted order.

    A = ∅;
    for(i = 1; i ≤ n; ++i){
        if(A ∪ {si} in F){
            A ← A ∪ {si};
        }
    }
    return A;
}
```

Recall the greedy algorithm

```
Greedy(M = (S, F), ω){

    Sort elements in S by their weights into the non-increasing order;
    // Assume that s1, s2, ..., sn is the sorted order.

    A = ∅;
    for(i = 1; i ≤ n; ++i){
        if(A ∪ {si} in F){
            A ← A ∪ {si};
        }
    }
    return A;
}
```

The returned subset $A = \{a_1, a_2, \ldots, a_x\}$ must be in F.
Let $r(a_1) < r(a_2) < \ldots < r(a_x)$,
where $r(e)$ denotes the rank of e in the sorted S.

Proof (1/3)

<u>Claim 1</u>. A is maximal. In other words, A is not a subset of any other set in F.

Proof (1/3)

Claim 1. A is maximal. In other words, A is not a subset of any other set in F.

Proof. Suppose that $A \subset B$ for some $B \in F$ and $|B| > |A|$. Let z be some element in $B \setminus A$ so that $A \cup \{z\}$ in F. Such z exists due to the exchange property. By the hereditary property, $\{a_1, a_2, \ldots, a_t\} \cup \{z\}$ in F for any $t \in [1, x]$.

If $r(a_x) < r(z)$, then Greedy should have added some e into A so that $r(a_x) < r(e) \leq r(z)$. $\rightarrow\leftarrow$

Otherwise $r(a_{t-1}) < r(z) < r(a_t)$ for some $t \in [1, x]$, then Greedy should have added some e into A so that $r(a_{t-1}) < r(e) \leq r(z)$. $\rightarrow\leftarrow$

Proof (2/3)

Claim 2. Every maximal subset in F has the same size.

Proof. Let A and B be two maximal subsets in F and $|A| < |B|$.
By the exchange property, there exists some z in $B \setminus A$ so that $A \cup \{z\}$ in F, contradicting the maximality of A.

Proof (2/3)

Claim 2. Every maximal subset in F has the same size.

Proof. Let A and B be two maximal subsets in F and $|A| < |B|$.
By the exchange property, there exists some z in B \ A so that $A \cup \{z\}$ in F, contradicting the maximality of A.

Claim 3. Some maximal subset in F is a max-weight subset in F.

Proof. Let A be a max-weight subset but A is a proper subset of some B in F. Then, B also has the max-weight because the weight function w is non-negative. Such a B is a witness.

Proof (2/3)

Claim 2. Every maximal subset in F has the same size.

Proof. Let A and B be two maximal subsets in F and |A| < |B|.
By the exchange property, there exists some z in B \ A so that A ∪ {z} in F, contradicting the maximality of A.

Claim 3. Some maximal subset in F is a max-weight subset in F.

Proof. Let A be a max-weight subset but A is a proper subset of some B in F. Then, B also has the max-weight because the weight function w is non-negative. Such a B is a witness.

Both A and a max-weight subset are maximal.

Proof (3/3)

Let the max-weight subset be $B = \{b_1, b_2, \ldots, b_x\}$ where $r(b_1) < r(b_2) < \ldots < r(b_x)$. Since $A \neq B$, there exists some $t \in [1, x]$ so that $r(a_1) = r(b_1)$, $r(a_2) = r(b_2)$, …, $r(a_{t-1}) = r(b_{t-1})$, $r(a_t) < r(b_t)$. If there are multiple choices of max-weight susbets, we pick B as the one whose t is largest among all.

Proof (3/3)

Let the max-weight subset be $B = \{b_1, b_2, \ldots, b_x\}$ where $r(b_1) < r(b_2) < \ldots < r(b_x)$. Since $A \neq B$, there exists some $t \in [1, x]$ so that $r(a_1) = r(b_1)$, $r(a_2) = r(b_2)$, …, $r(a_{t-1}) = r(b_{t-1})$, $r(a_t) < r(b_t)$. If there are multiple choices of max-weight susbets, we pick B as the one whose t is largest among all.

Theorem. $A = B$, and therefore A is a max-weight subset.

Proof (3/3)

Let the max-weight subset be $B = \{b_1, b_2, \ldots, b_x\}$ where $r(b_1) < r(b_2) < \ldots < r(b_x)$. Since $A \neq B$, there exists some $t \in [1, x]$ so that $r(a_1) = r(b_1)$, $r(a_2) = r(b_2)$, …, $r(a_{t-1}) = r(b_{t-1})$, $r(a_t) < r(b_t)$. If there are multiple choices of max-weight susbets, we pick B as the one whose t is largest among all.

<u>Theorem</u>. A = B, and therefore A is a max-weight subset.

Proof. Let $A' = \{a_1, a_2, \ldots, a_t\}$. By exchange property, we could iteratively add some element from $B \setminus A'$ to A' until A' has the same size as B. Finally, A' becomes $B \cup \{a_t\} \setminus \{z\}$ for some z in $\{b_t, b_{t+1}, \ldots, b_x\}$. Since $r(a_t) < r(b_t) \leq r(z)$, $w(a_t) \geq w(z)$ and A' is another max-weight subset, violating the condition that B is the one whose t is largest.