MEASURE AND CONQUER

Shlomo Waksnine
agenda

- Recurrence equations
- Independent set
- Set cover
  - Dominating set
- Lower bounds
RECURRENCES
We define a linear homogenous recurrence to be a sequence of the form:

\[ T(n) = T(n - t_1) + T(n - t_2) + \cdots + T(n - t_r) \]

where \( 1 \leq t_i \leq n \).

The vector \( b = (t_1, \ldots, t_r) \) is called the **branching vector**.

The polynom \( p(x) = x^n - x^{n-t_1} - \cdots - x^{n-t_r} \) is called the characteristic polynomial.

Thm: \( T(n) = O^*(\alpha^n) \) where \( \alpha \) is the unique positive real root of \( p \).

\( \alpha \) is called the **branching factor**.
Let $T(n) = T(n - 1) + T(n - 2)$.
The branching vector is $(1, 2)$.
The characteristic polynomial is $x^2 - x - 1$.
The roots are: $\frac{1+\sqrt{5}}{2}, \frac{1-\sqrt{5}}{2}$.
The branching factor is $\frac{1+\sqrt{5}}{2}$.
Thus, $T(n) = O^*\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.
measure and conquer

- Every algorithm that uses recursion induces a branching vector for each branch.
- The algorithm induces a tree of its execution.
- Finding the number of leaves is sufficient to find the running time.
- Solving each branch recurrence and take the maximum gives us upper bound.
- We used to analyze algorithms based on a “natural” parameter of the input.
  - Number of vertices in a graph.
- Using a different measure of the input can give better results.
Independent set

- Given a graph $G = (V, E)$, a set $I \subseteq V$ is called independent if there is no edge between each pair in $I$.
- Our goal is to find the independent set of maximum size.
Every singleton vertex must be in the maximum independent set.
If \( v \) has only one neighbor \( w \), then \( v \) belongs to some maximum independent set.
If $\Delta(G) \leq 2$ then we can find the maximum independent set in polynomial time. 

$G$ is composed of trees and cycles.
Independent set - algorithm

1. If \( \exists v \in V \) with \( d(v) = 0 \) then:
   
   return \( 1 + mis(G \setminus \{v\}) \)

2. If \( \exists v \in V \) with \( d(v) = 1 \) then:
   
   return \( 1 + mis(G \setminus N[v]) \)

3. If \( \Delta(G) \geq 3 \) then:
   
   choose a vertex \( v \) of maximum degree in \( G \)
   
   return \( \max(1 + mis(G \setminus N[v]), mis(G \setminus \{v\})) \)

4. If \( \Delta(G) \leq 2 \) then:
   
   return maximum independent set of \( G \) using polynomial time algorithm
The algorithm has only one branch on step 3.

Branching on $v$ means that $d(v) \geq 3$.

Removing $v$ – remove 1 from the original problem size.

Add $v$ to the independent set – remove at least 4 from the original problem size.

3. If $\Delta(G) \geq 3$ then:
   - choose a vertex $v$ of maximum degree in $G$
   - return $\max(1 + mis(G \setminus N[v]), mis(G \setminus \{v\}))$
simple analysis – cont.

- \( T(n) \leq T(n - 1) + T(n - 4) \).
- The characteristic polynomial is \( x^4 - x^3 - 1 \).
- The branching vector is (1,4).
- The roots are: \(-0.82, 1.3803, 0.22 + 0.91i, 0.22 - 0.91i\).
- Thus the branching factor is \( \tau(1,4) = 1.3803 \).
- \( T(n) = O^*(1.3803^n) \).
- Can we do better?
better analysis

- $n_i$ - number of vertices with degree $i$.
- Assign weight $w_2$ to vertices with degree 2.
- Analyze the running time according to the measure:
  \[ k(G) = w_2 n_2 + n_{\geq 3} \]
- We analyze the running time with respect to $w_2 = 0.5$. 
We have only one branching step.

Suppose the algorithm branches on a vertex $v$ of degree $d \geq 3$:

- **OUT** – the decrease of the measure by discarding $v$.
- **IN** – the decrease of the measure by adding $v$ to the independent set.
intuition - IN case

$n_2 = 4$

$n_3 = 2$

$n_2 = 0$

$n_3 = 0$
analysis - cont.

- Removing $v$ decreases the measure by 1 in both cases.
- Let $u_1, \ldots, u_d$ be the neighbors of $v$.
- If $d(u_i) = 2$ then the measure is decreased by 0.5 in both cases.
- If $d(u_i) \geq 3$ then we decrease the measure by 1 in case IN.
- Thus, $IN + OUT \geq 2 + d(v)$.
- Thus, $\tau(IN,OUT) \leq \tau(1,1 + d(v))$.
- If $d(v) \geq 4$, then $\tau(IN,OUT) \leq \tau(1,5) < 1.3248$. 
analysis – cont.

- Suppose we branch on $v$ with $d(v) = 3$.
- Each vertex in $G$ has degree 2 or 3.
- Removing $v$ decreases the measure by 1 in both cases.
- OUT – remove $v$ decreases the weight of its neighbors by 0.5.
- IN – remove the neighbors of $v$ decreases the measure by 0.5 or 1.
- Thus, $OUT, IN \geq 1 + 3 \cdot 0.5 = 2.5$.
- Thus, $\tau(OUT, IN) \leq \tau(2.5,2.5) < 1.3196$.
- The running time of the algorithm is $O^*(1.3248^n)$. 
SET COVER
set cover

- $\mathcal{U}$ – set of elements (universe).
- $\mathcal{S}$ – collection of (non-empty) subsets of $\mathcal{U}$.
- A set cover of $(\mathcal{U}, \mathcal{S})$ is a subset $\mathcal{S}' \subseteq \mathcal{S}$ which covers $\mathcal{U}$:
  \[ \bigcup_{S \in \mathcal{S}'} S = \mathcal{U} \]
- Our goal is to find a set cover $\mathcal{S}'$ with minimum cardinality.
- We assume that $\mathcal{S}$ covers $\mathcal{U}$. 
set cover - example

- \{S_1, S_4, S_5, S_6\} is a cover
- \{S_3, S_4, S_5\} is a minimum cover
If there are two distinct sets $S$ and $R$ in $\mathcal{S}$, $S \subseteq R$, then there is a minimum set cover which does not contain $S$.

Thus:

$$\text{msc}(\mathcal{S}) = \text{msc}(\mathcal{S} \setminus \{S\})$$
If there is an element \( u \) of \( U \) which belongs to a unique \( S \in \mathcal{S} \), then \( S \) belongs to every set cover.

Thus:

\[
\text{msc}(\mathcal{S}) = 1 + \text{msc}(\text{del}(S, \mathcal{S}))
\]

where:

\[
\text{del}(S, \mathcal{S}) = \{Z \mid Z = R \setminus S \neq \emptyset, R \in \mathcal{S}\}
\]
For a given MSC instance $S$ such that all the subsets $S$ of $S$ are of cardinality two, MSC can be solved in polynomial time.

- Reduction to maximum matching.

$\mathcal{U} = \{1, 2, 3, 4, 5\}$

$\mathcal{S} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 5\}, \{4, 5\}\}$

$MSC(S) = \{\{1, 3\}, \{2, 4\}, \{4, 5\}\}$ or $MSC(S) = \{\{1, 3\}, \{2, 4\}, \{3, 5\}\}$
set cover - algorithm

Algorithm $\text{msc}(S)$.

Input: A collection $S$ of subsets of a universe $U$.
Output: The minimum cardinality of a set cover of $S$.

1. if $|S| = 0$ then
   return 0
2. if $\exists S, R \in S$ with $S \subset R$ then
   return $\text{msc}(S \backslash \{S\})$
3. if $\exists u \in U(S)$ such that there is a unique $S \in S$ with $u \in S$ then
   return $1 + \text{msc}(\text{del}(S, S))$
4. choose a set $S \in S$ of maximum cardinality
5. if $|S| = 2$ then
   return $\text{poly-msc}(S)$
6. if $|S| \geq 3$ then
   return $\min(\text{msc}(S \backslash \{S\}), 1 + \text{msc}(\text{del}(S, S)))$

Fig. 6.4 Algorithm $\text{msc}$ for MSC
set cover - analysis

- $n_i$ - number of subsets $S \in \mathcal{S}$ of cardinality $i$.
- $m_j$ - number of elements $u \in U$ with frequency $j$.
- We analyze the running with respect to the following measure:

$$k(S) = \sum_{i \geq 1} w_i n_i + \sum_{j} v_j m_j$$
To simplify the analysis we make the following assumptions:

- $w_i \leq w_{i+1}$
- $v_j \leq v_{j+1}$
- $w_1 = v_1 = 0$
- $w_i = v_i = 1$ for $i \geq 6$.
- $\Delta(w_i) \geq \Delta(w_{i+1})$ for $i \geq 2$ ($\Delta(w_i) = w_{i+1} - w_i$).
Let $\ell(k)$ be the number of leaves in the search tree generated by the algorithm to solve a problem of measure $k$.

- Conditions 2 and 3 implies $\ell(k) \leq \ell(k - w_{|S|})$.
- Condition 5 implies that $\ell(k) = 1$.
- Condition 6 implies two subproblems:
  - $S_{OUT} = S \setminus S$
  - $S_{IN} = \text{del}(S,S)$

$$k(S) = \sum_{i \geq 1} w_i n_i + \sum_j v_j m_j$$
analyze $S_{IN} = del(S, S)$

- Removing $S$ implies decreasing by $w_{|S|}$.
- $r_{\geq i} = \sum_{j \geq i} r_j$ - number of elements of $S$ of frequency at least $i$.
- Removing $u \in S$ with frequency $i$ implies decreasing by $v_i$.
- Overall: $\sum_{i \geq 2} r_i v_i = \sum_{i=2}^5 r_i v_i + r_{\geq 6}$

$$k(S) = \sum_{i \geq 1} w_i n_i + \sum_j v_j m_j$$
analyze $S_{IN} = del(S, S)$

- Let $R$ be a set sharing an element $u$ with $S$.
  - $|R| \leq |S|$
  - By removing $u$, the cardinality of $R$ is reduced by one.
  - Hence, implies a reduction of the size of $S_{IN}$ by $\Delta w_{|R|} \geq \Delta w_{|S|}$.

- Overall:
  $$\Delta w_{|S|} \sum_{i \geq 2} (i - 1) r_i \geq \Delta w_{|S|} \left( \sum_{i=2}^{6} (i - 1) r_i + 6 \cdot r_{\geq 7} \right)$$

- Hence, $\Delta k_{IN} \geq w_{|S|} + \sum_{i=2}^{5} r_i v_i + r_{\geq 6} + \Delta w_{|S|} \left( \sum_{i=2}^{6} (i - 1) r_i + 6 \cdot r_{\geq 7} \right)$

$k(S) = \sum_{i \geq 1} w_i n_i + \sum_{j} v_j m_j$
analyze $S_{OUT} = S \setminus S$

- The overall decrease in the measure is:

$$\Delta k_{OUT} \geq w_{|S|} + \sum_{i=2}^{6} r_i \Delta v_i + \Delta k'$$

- Where:

$$\Delta k' = \begin{cases} 
0, & r_2 = 0 \\
v_2 + w_2, & r_2 = 1 \\
\begin{array}{l}
v_2 + \min\{2w_2, w_3\} = w_3, \\
v_2 + \min\{3w_2, w_2 + w_3\} = w_2 + w_3, \\
v_2 + \min\{3w_2, w_2 + w_3, w_4\} = w_4,
\end{array} & r_2 = 2, |S| = 3 \\
\begin{array}{l}
\min\{2w_2, w_3, w_4\}, \\
\min\{3w_2, w_2 + w_3, w_4\}, \\
2w_2 + w_3, \\
3w_2, \\
w_2 + w_3, \\
w_2 + w_3 + w_4, \\
w_2 + w_3 + w_4 + w_5
\end{array} & r_2 \geq 3, |S| \geq 4
\end{cases}$$
Notice that $\Delta w_{|S|} = 0$ for $|S| \geq 7$.

Hence it is sufficient to restrict ourselves to the recurrences for the cases $3 \leq |S| \leq 7$.

Need to find optimal $(w_2, w_3, w_4, w_5, v_2, v_3, v_4, v_5)$.

For each combination of the values of $|S|$ and $r_2, ..., r_{|S|}$.

Which yields branching factor $\alpha < 1.2353$.

Thus, the overall running time is $O^*(\ell(k)) = O^*(1.2353^{|U|+|S|})$. 

\begin{align*}
w_i &= \begin{cases} 
0.377443 & \text{if } i = 2, \\
0.754886 & \text{if } i = 3, \\
0.909444 & \text{if } i = 4, \\
0.976388 & \text{if } i = 5, 
\end{cases} \\
\text{and } v_i &= \begin{cases} 
0.399418 & \text{if } i = 2, \\
0.767579 & \text{if } i = 3, \\
0.929850 & \text{if } i = 4, \\
0.985614 & \text{if } i = 5, 
\end{cases}
\end{align*}
from set cover to dominating set

\[ \mathcal{U} = \{1, 2, 3, 4, 5\} \]
\[ \mathcal{S} = \{\{1, 2, 3\}, \{2, 1, 3, 4\}, \{3, 1, 2, 5\}, \{4, 2, 5\}, \{5, 3, 4\}\} \]
from set cover to dominating set

- $D$ is a dominating set of $G$ if and only if $\{N[v] | v \in D\}$ is a set cover of $\{N[v] | v \in V\}$.
- $|U| = |S| = |V| = n$
- Thus, we can solve minimum dominating set problem with $O^*(1.2353^{2n}) = O^*(1.5259^n)$
LOWER BOUNDS
lower bounds

- What we did is to find upper bound on the running time of the algorithm.
- It is useful to find a lower bound.
  - Tells us how good is our analysis.
independent set

1. If \( \exists v \in V \) with \( d(v) = 0 \) then:
   return \( 1 + mis(G \setminus \{v\}) \)

2. If \( \exists v \in V \) with \( d(v) = 1 \) then:
   return \( 1 + mis(G \setminus N[v]) \)

3. If \( \Delta(G) \geq 3 \) then:
   choose a vertex \( v \) of maximum degree in \( G \)
   return \( \max(1 + mis(G \setminus N[v]), mis(G \setminus \{v\})) \)

4. If \( \Delta(G) \leq 2 \) then:
   return maximum independent set of \( G \) using polynomial time algorithm
independent set

Define $G_n = (V, E)$ such that $V = [n]$ and $(i, j) \in E$ if and only if $|i - j| = 2$. 
G₇ execution

- Add 3 to the independent set induces subproblem of the form $G_{n-5}$.
- Removing 3 induces subproblem of the form $G_{n-3}$.
- $T(n) = T(n - 5) + T(n - 3)$.
- The running time is $\Omega(1.19386^n)$. 