MEASURE AND CONQUER

Shlomo Waknine

agenda

- ❑ Recurrence equations
- ❑ Independent set
- ❑ Set cover
  - ❑ Dominating set
- ❑ Lower bounds

RECURRENCES

definitions

- 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 α is the unique positive real root of p.
- α is called the **branching factor**.

example

- 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

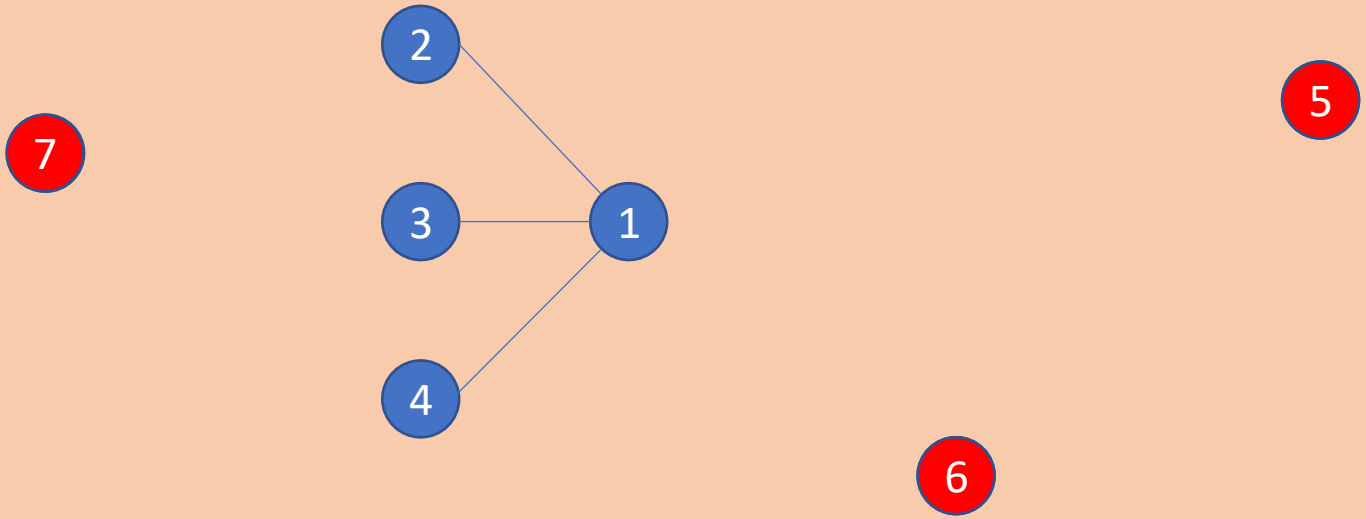
- Every algorithm that uses recursion induces branching vector for each branch.
- The algorithm induces a tree of its execution.
- Finding the number of leaves is sufficient to find the running time.
- Solving each branch recurrence and take the maximum gives us upper bound.
- We used to analyze algorithms 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.

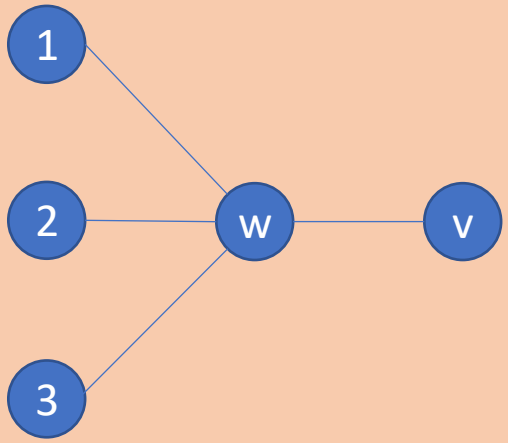
singletons

- Every singleton vertex must be in the maximum independent set.



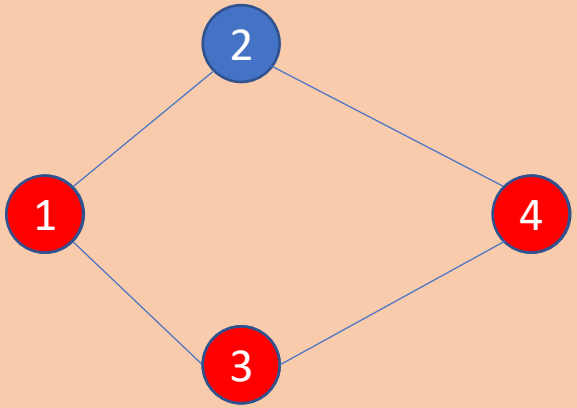
$d(v) = 1$

- If v has only one neighbor w, then v belongs to some maximum independent set.



$\Delta(G) \leq 2$

- 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 $\max(1 + mis(G \setminus N[v]), mis(G \setminus \{v\}))$

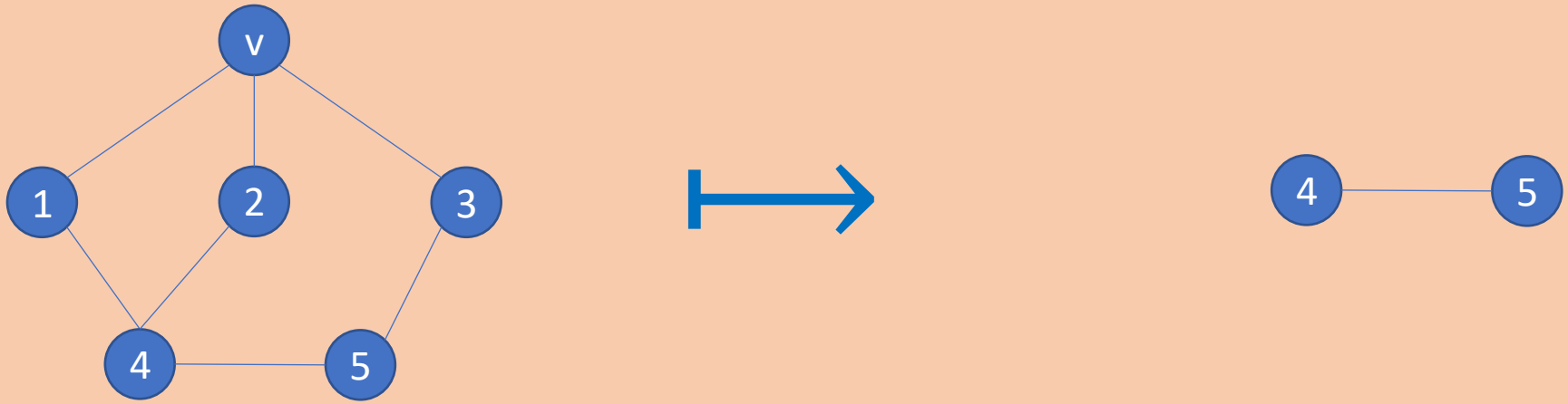
4. If $\Delta(G) \leq 2$ then:

   return maximum independent set of G using polynomial time algorithm

simple analysis - cont.

- 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.

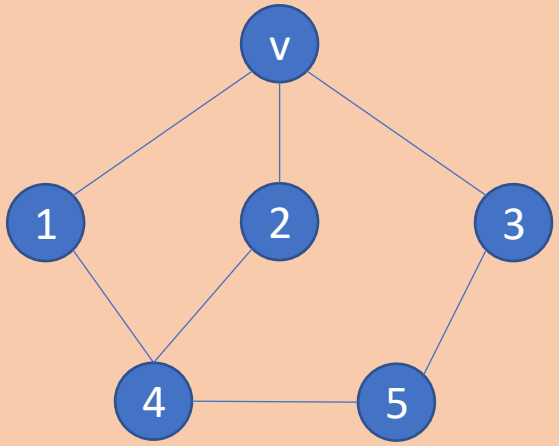
- $T(n) \leq T(n-1) + T(n-4)$.
- The characteristic polynomial is $x^4 - x^3 - 1$.
- The branching vector is $(1,4)$.
- The roots are: $-0.82, 1.3803, 0.22 + 0.91i, 0.22 - 0.91i$.
- Thus the branching factor is $\tau(1,4) = 1.3803$.
- $T(n) = O^*(1.3803^n)$.
- Can we do better?

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:

$$k(G) = w_2 n_2 + n_{\geq 3}$$

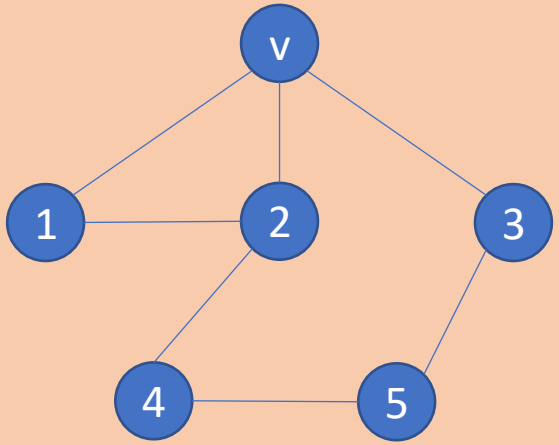
- We analyze the running time with respect to $w_2 = 0.5$.



analysis - cont.

- 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



$\longmapsto$

$n_2 = 4$

$n_3 = 2$

$n_2 = 0$

$n_3 = 0$

analysis - cont.

- Removing v decreases the measure by 1 in both cases.
- Let $u_1, \dots, u_d$ 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, $\tau(IN, OUT) \leq \tau\big(1{,}1 + d(v)\big)$.
- 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

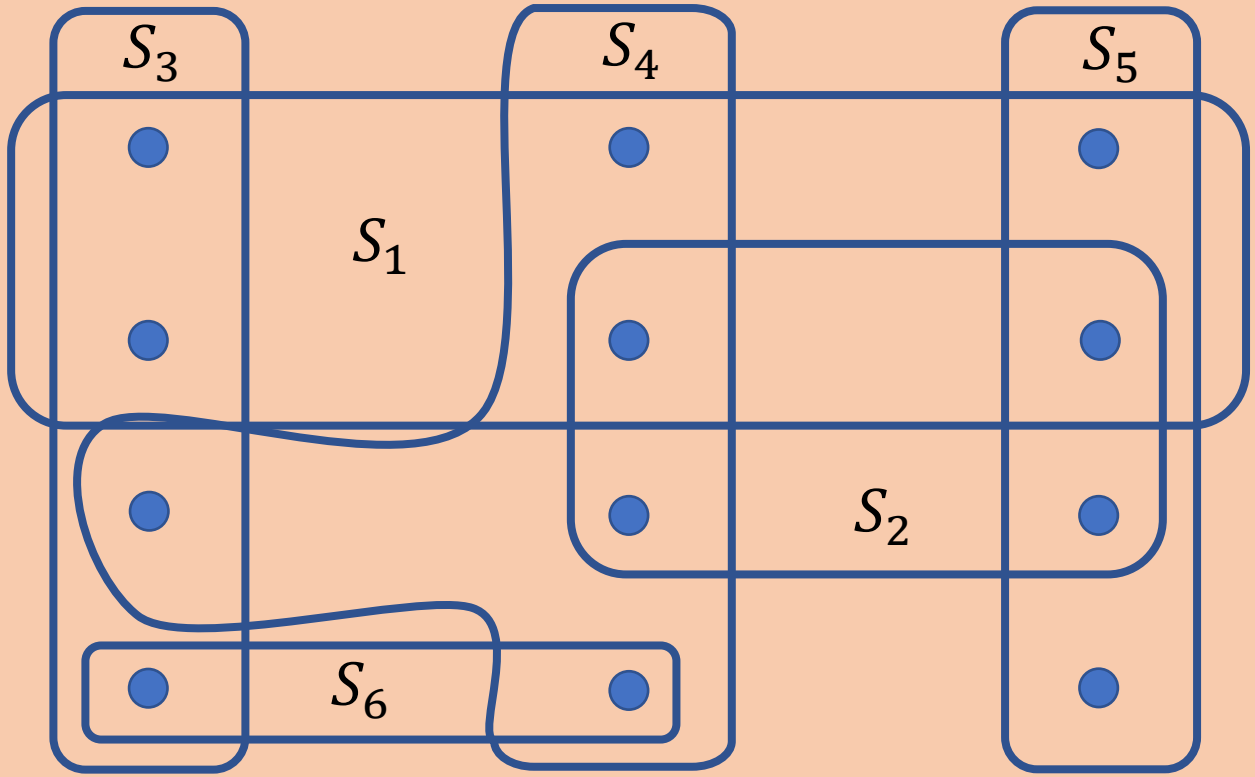
- $\mathcal{U}$ – set of elements (universe).
- $\mathcal{S}$ – collection of (non-empty) subsets of $\mathcal{U}$.
- A set cover of $(\mathcal{U}, \mathcal{S})$ is a subset $\mathcal{S}' \subseteq \mathcal{S}$ which covers $\mathcal{U}$:

$$\bigcup_{S \in \mathcal{S}'} S = \mathcal{U}$$

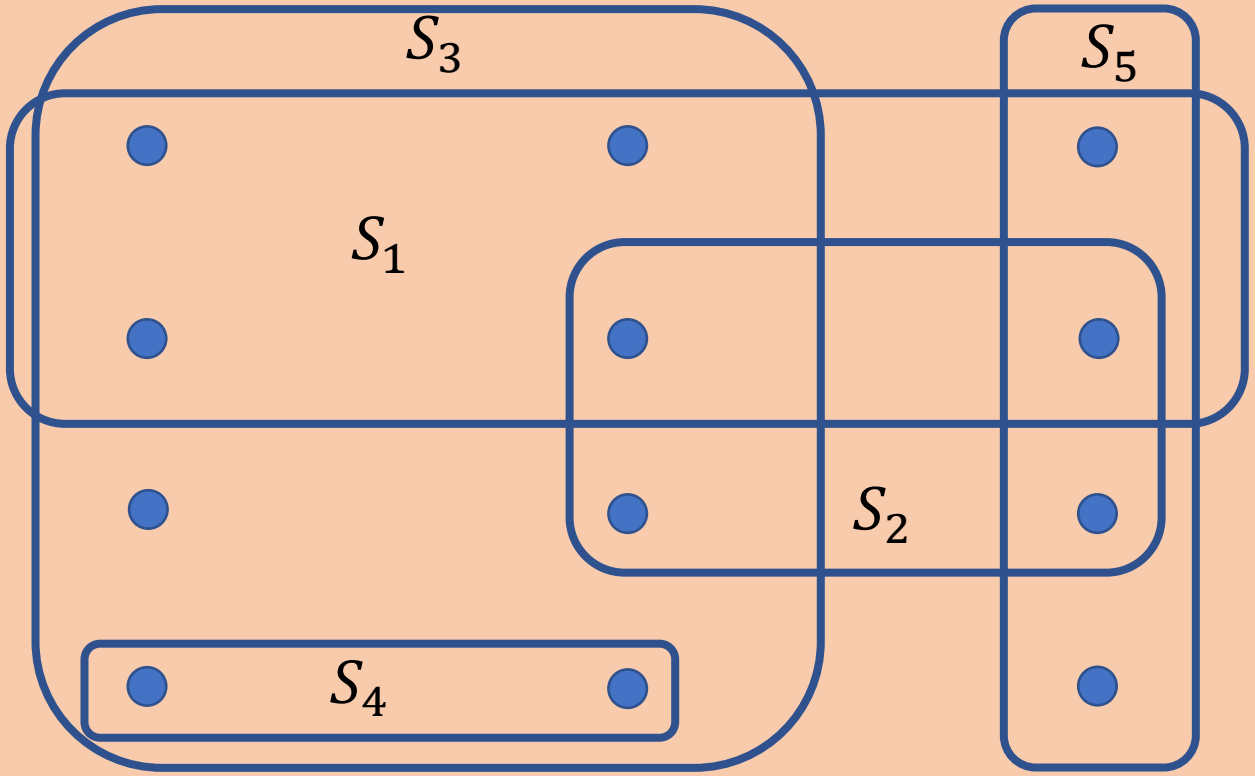
- Our goal is to find a set cover $\mathcal{S}'$ with minimum cardinality.
- We assume that $\mathcal{S}$ covers $\mathcal{U}$.

set cover - example

- $\{S_1, S_4, S_5, S_6\}$ is a cover
- $\{S_3, S_4, S_5\}$ is a **minimum** cover



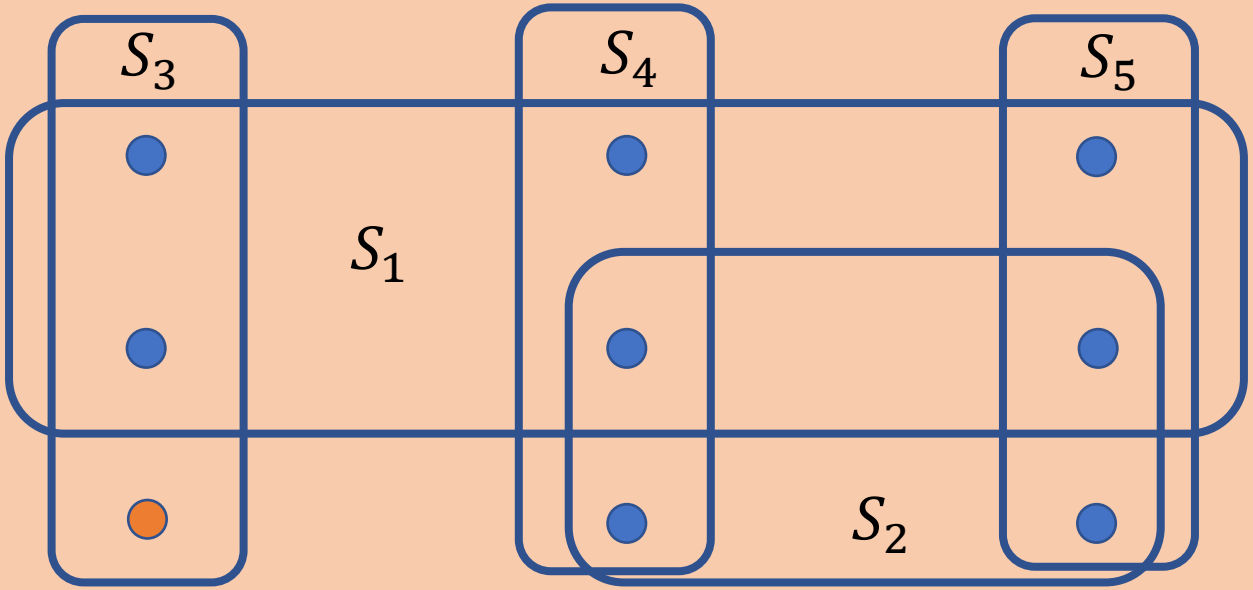
set cover - lemma 1



- 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:

$$\text{msc}(\mathcal{S})=\text{msc}(\mathcal{S} \setminus \{S\})$$

set cover - lemma 2



- 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:

$$msc(\mathcal{S}) = 1+msc(del(S,\mathcal{S}))$$

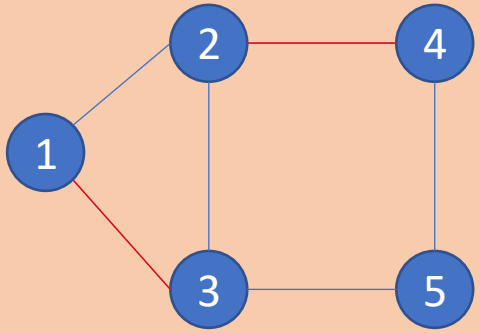
where:

$$del(S,\mathcal{S}) = \{Z \mid Z = R\backslash S \neq \phi, R \in \mathcal{S}\}$$

set cover - lemma 3

- For a given MSC instance S such that all the subsets S of $\mathcal{S}$ are of cardinality two, MSC can be solved in polynomial time.
- Reduction to maximum matching.

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



$MSC(\mathcal{S}) = \{\{1,3\},\{2,4\},\{4,5\}\}$ or $MSC(\mathcal{S}) = \{\{1,3\},\{2,4\},\{3,5\}\}$

set cover - algorithm

Algorithm msc($\mathcal{S}$).
Input: A collection $\mathcal{S}$ of subsets of a universe $\mathcal{U}$.
Output: The minimum cardinality of a set cover of $\mathcal{S}$.

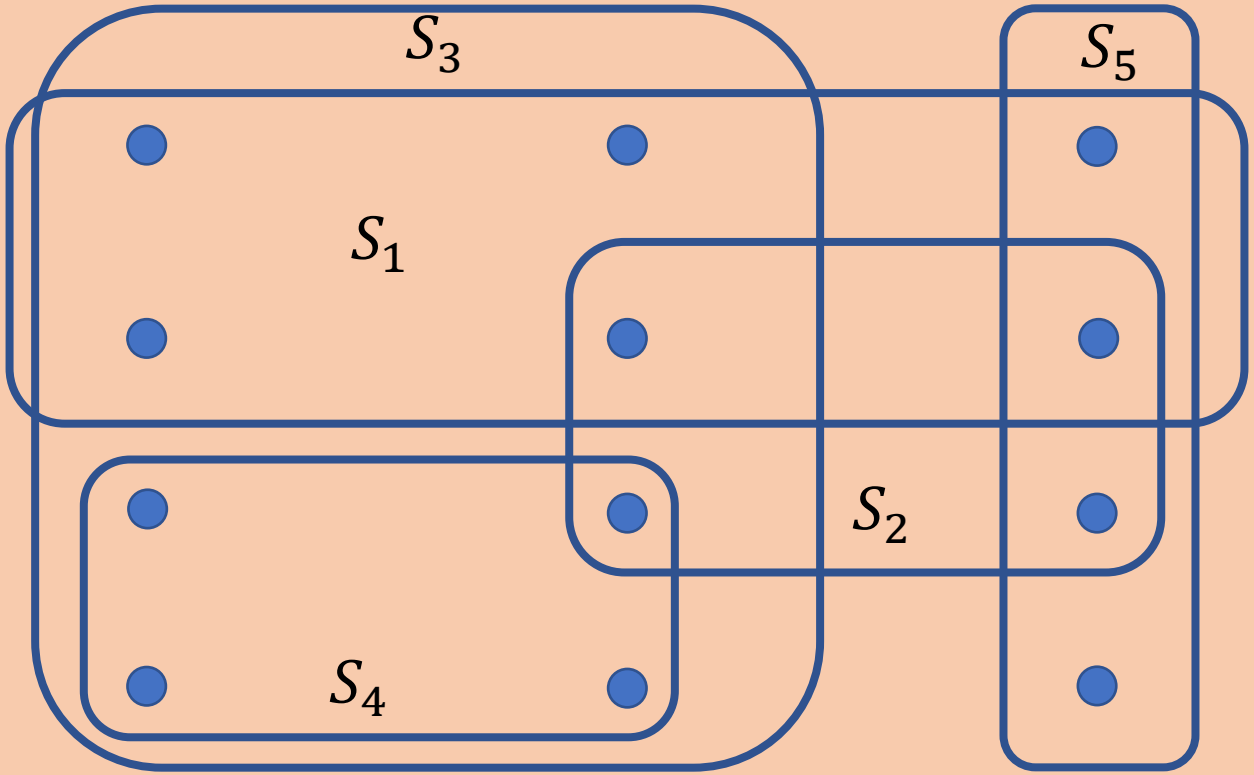
if $|\mathcal{S}| = 0$ **then**
  return 0
if $\exists S, R \in \mathcal{S}$ with $S \subseteq R$ **then**
  return $\text{msc}(\mathcal{S} \setminus \{S\})$
if $\exists u \in \mathcal{U}(\mathcal{S})$ such that there is a unique $S \in \mathcal{S}$ with $u \in S$ **then**
  return $1 + \texttt{msc}(\text{del}(S, \mathcal{S}))$
choose a set $S \in \mathcal{S}$ of maximum cardinality
if $|S| = 2$ **then**
  return $\texttt{poly-msc}(\mathcal{S})$
if $|S| \geq 3$ **then**
  return $\min(\texttt{msc}(\mathcal{S} \setminus \{S\}), 1 + \texttt{msc}(\text{del}(S, \mathcal{S})))$

Fig. 6.4 Algorithm `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:

$$k(\mathcal{S}) = \sum_{i \geq 1} w_i n_i + \sum_j v_j m_j$$



analysis - cont.

- 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 $i \geq 2$ ($\Delta(w_i) = w_{i+1} - w_i$).

analysis - cont.

- 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 $\ell(k) \leq \ell\left(k - w_{|S|}\right)$.
- Condition 5 implies that $\ell(k) = 1$.
- Condition 6 implies two subproblems:
  - $\mathcal{S}_{OUT} = \mathcal{S} \setminus S$
  - $\mathcal{S}_{IN} = del(S, \mathcal{S})$

Algorithm msc($\mathcal{S}$).
Input: A collection $\mathcal{S}$ of subsets of a universe $\mathcal{U}$.
Output: The minimum cardinality of a set cover of $\mathcal{S}$.

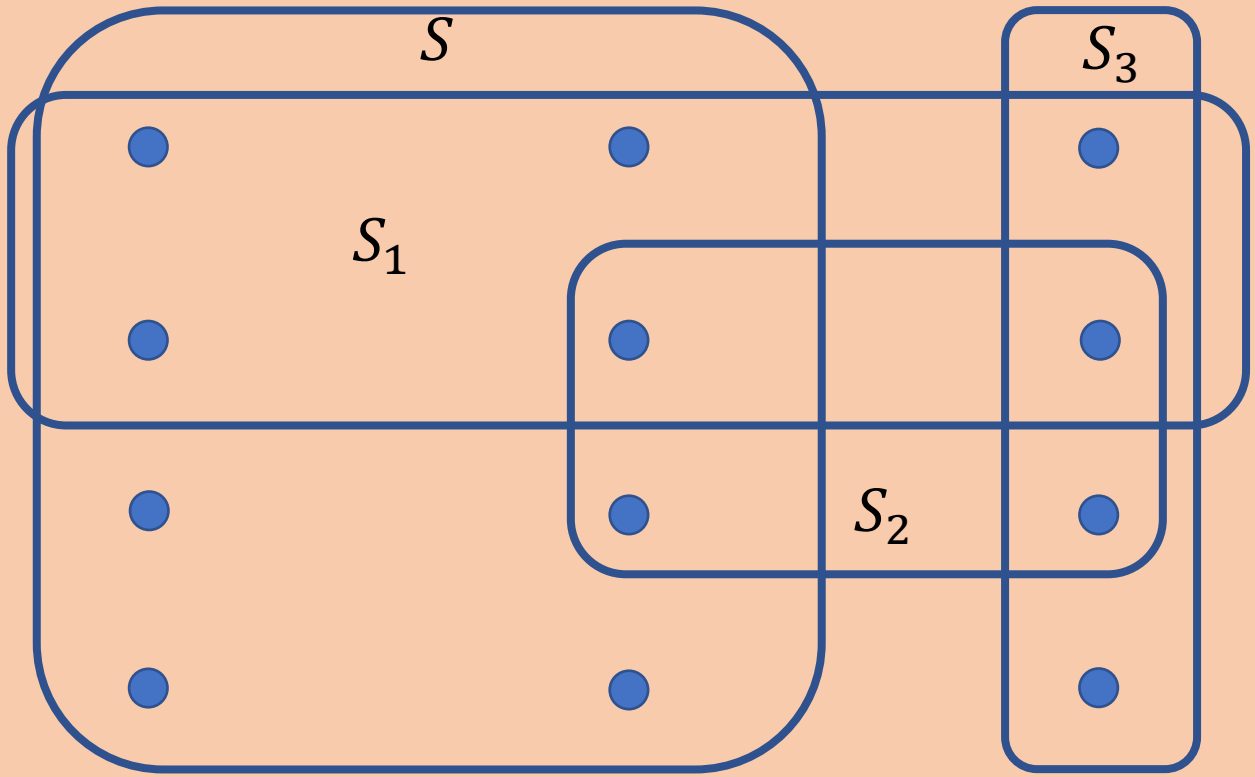
1 **if** $|\mathcal{S}| = 0$ **then**
  return 0
2 **if** $\exists S, R \in \mathcal{S}$ with $S \subseteq R$ **then**
  return msc($\mathcal{S} \setminus \{S\}$)
3 **if** $\exists u \in \mathcal{U}(\mathcal{S})$ such that there is a unique $S \in \mathcal{S}$ with $u \in S$ **then**
  return $1 +$ msc(del($S,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

Fig. 6.4 Algorithm msc for MSC

$$k(\mathcal{S}) = \sum_{i \geq 1} w_i n_i + \sum_{j} v_j m_j$$

analyze $\mathcal{S}_{IN} = del(S, \mathcal{S})$

- 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: $\sum_{i \geq 2} r_i v_i = \sum_{i=2}^{5} r_i v_i + r_{\geq 6}$



$$k(\mathcal{S}) = \sum_{i \geq 1} w_i n_i + \sum_j v_j m_j$$

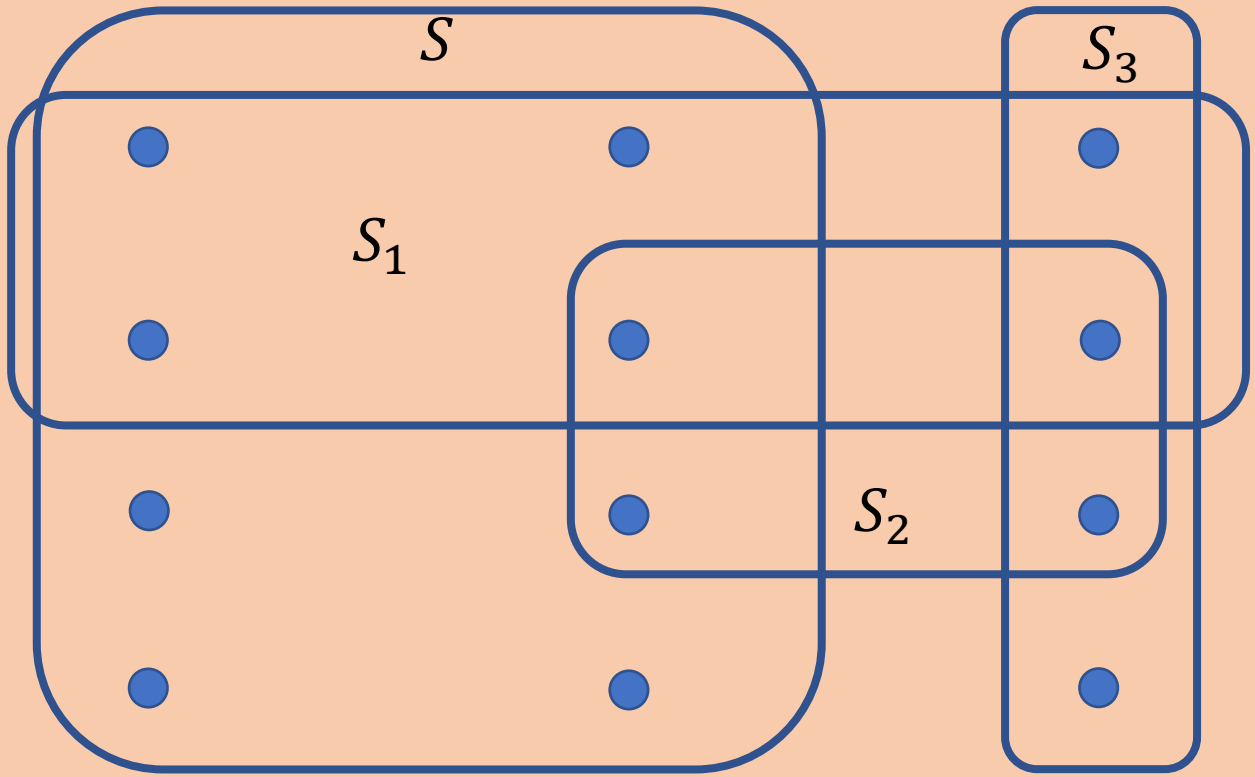
analyze $\mathcal{S}_{IN} = del(S, \mathcal{S})$

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



$$k(\mathcal{S}) = \sum_{i \geq 1} w_i n_i + \sum_{j} v_j m_j$$

analyze $S_{OUT} = \mathcal{S} \setminus S$

- 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:

$$\Delta k' = \begin{cases} 0, & r_2 = 0 \\ v_2 + w_2, & r_2 = 1 \\ v_2 + \min\{2w_2, w_3\} = w_3, & r_2 = 2 \\ v_2 + \min\{3w_2, w_2 + w_3\} = w_2 + w_3, & r_2 = 3, |S| = 3 \\ v_2 + \min\{3w_2, w_2 + w_3, w_4\} = w_4, & r_2 \geq 3, |S| \geq 4 \end{cases}$$

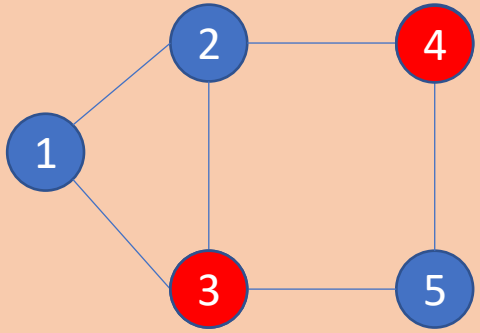
analysis - end

- 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 $(w2, w3, w4, w5, v2, v3, v4, v5)$.
  - For each combination of the values of $|S|$ and $r_2, \ldots, r_{|S|}$.

$$w_i = \begin{cases} 0.377443 & \text{if } i = 2, \\ 0.754886 & \text{if } i = 3, \\ 0.909444 & \text{if } i = 4, \\ 0.976388 & \text{if } i = 5, \end{cases} \quad \text{and} \quad v_i = \begin{cases} 0.399418 & \text{if } i = 2, \\ 0.767579 & \text{if } i = 3, \\ 0.929850 & \text{if } i = 4, \\ 0.985614 & \text{if } i = 5, \end{cases}$$

  - Which yields branching factor $\alpha < 1.2353$.
- Thus, the overall running time is $O^*\big(\ell(k)\big) = O^*\big(1.2353^{|\mathcal{U}|+|\mathcal{S}|}\big)$.

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 $\{N[v] | \ v \in V\}$.
  - $|\mathcal{U}| = |\mathcal{S}| = |V| = n$
- 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 $\max(1 + mis(G \setminus N[v]), mis(G \setminus \{v\}))$

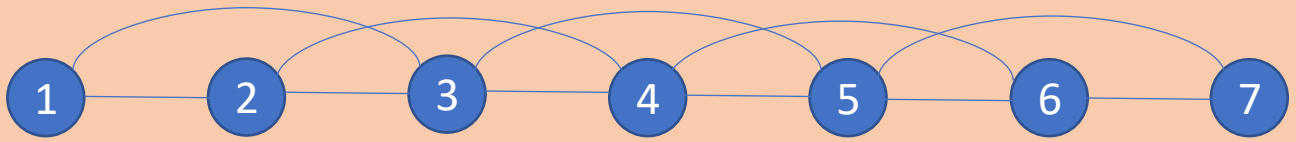
4. If $\Delta(G) \leq 2$ then:

   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 $|i - j| = 2$.

G_7 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 $\Omega(1.19386^n)$.