طراحی الگوریتم

درس هشتم : مسائل P و NP

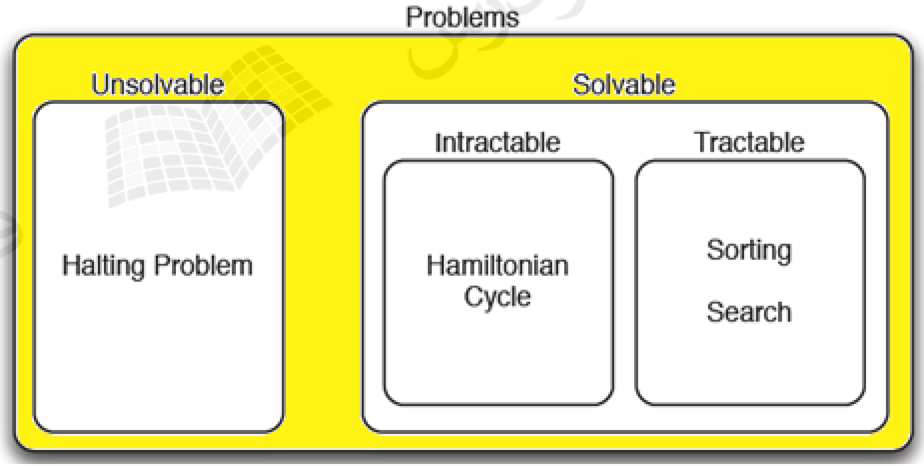
مدرس: فرشید شیرافکن

دانشجوی دکتری دانشگاه تهران
(کارشناسی و کارشناسی ارشد : کامپیوتر نرم افزار)
(دکتری: بیو انفورماتیک)

دسته بندی مسائل

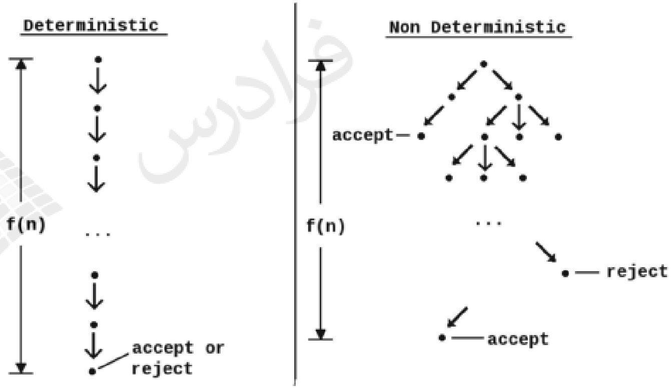
1- Unsolvable

2- Solvable:
- a - Intractable
- b - Tractable



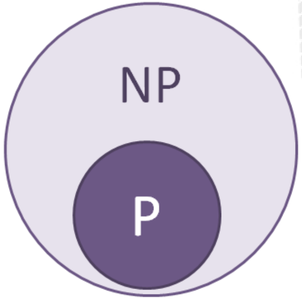
الگوریتم های قطعی و غیر قطعی

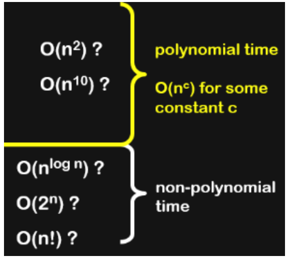
1 - Deterministic Algorithm
2 - Nondeterministic Algorithm



P: The class of problems, for which a deterministic Polynomial-time algorithm exists.

NP: The class of problems, for which a Nondeterministic Polynomial-time algorithm exists.

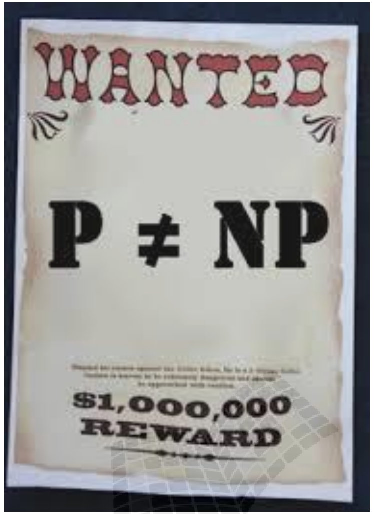


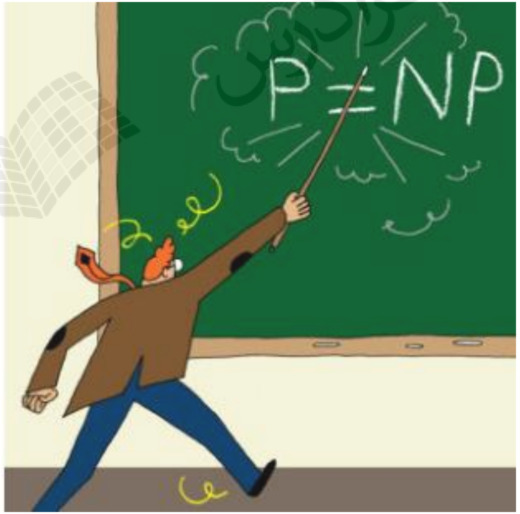


Nondeterministic algorithms
produce an answer by a series of "correct guesses"

Deterministic algorithms
make decisions based on information. (like those that a computer executes)

آیا P با NP برابر است؟





بیان الگوریتم های غیر قطعی

برای بیان الگوریتم های غیر قطعی توابع زیر را معرفی می کنیم:

choose(S) : Concurrently create S copies of the machine and assign one element of S to each copy.

success( ) : Halt the machine with finding a solution.

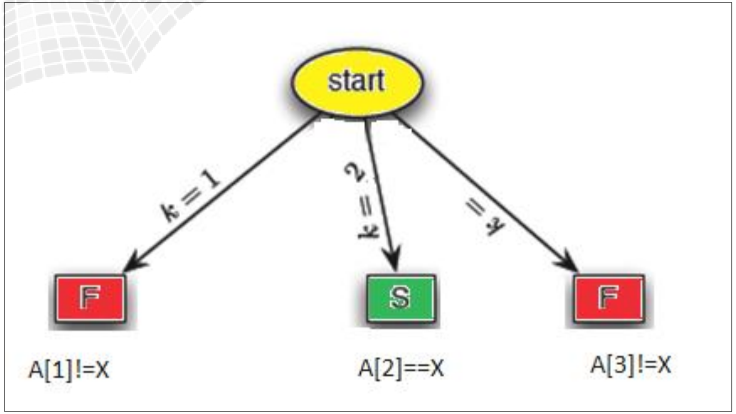
failure( ) : Halt the machine without finding a solution.

پیچیدگی زمانی هر سه تابع برابر O(1) است.

جستجوی غیر قطعی

```
NSearch (A[1···n], x)
k ← choose({1, 2, ··· , n}) ;
if A[k] = x then
    success() ;
else
    failure() ;
end if
end.
```

A[1..3]={7,5,9}
x=5



مرتبه O(1)

بهینه سازی – تصمیم گیری

Optimization vs. Decision

Any optimization problem can be converted to the corresponding decision problem by adding a bounding value.

جواب مسائل تصمیم گیری yes یا no می باشند.

مثال (بهینه سازی – تصمیم گیری)

بهینه سازی:

Find an $X = (x_1, x_2, \cdots, x_n)$ that maximize $\sum_{i=1}^{n} x_i p_i$ with respect to $\sum_{i=1}^{n} x_i w_i \leq b$.

تصمیم گیری: آیا جوابی با سود بیشتر از **k** وجود دارد یا نه؟

For a given k, is there a feasible solution, say $X = (x_1, x_2, \cdots, x_n)$, where $\sum_{i=1}^{n} x_i p_i \geq k$?

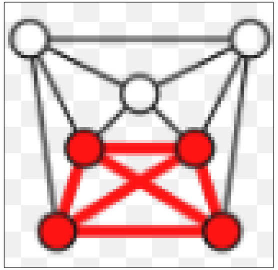
مثال (بهینه سازی – تصمیم گیری)

مسئله Max -Clique

- Optimization: Find a maximal subset $V' \subseteq V$, such that the induced graph by V' is complete graph

- Decision: For a given k, is there a subset $V' \subseteq V$ with $|V'| \geq k$, such that the induced graph by V' is complete graph (a clique of size at least k)?

کلیک : زیر گراف کامل



ارتباط بین مسائل بهینه سازی و مسائل تصمیم گیری

مسائل **Decision** چون جواب **yes** و **no** می دهند را می توان با هم مقایسه کرد.
مقایسه مسائل **optimization** ساده نیست چون مثلا جواب یک مسئله عدد است و مسئله دیگر بردار.

اگر حل مسئله بهینه سازی ساده باشد، حل مسئله تصمیم گیری معادل آن هم ساده خواهد بود.
اگر حل مسئله تصمیم گیری سخت باشد، آنگاه حل مسئله بهینه سازی معادل آن هم سخت خواهد بود.

بهینه سازی می تواند سخت باشد، اما تصمیم گیری ساده باشد.

برای اینکه نشان دهیم مسئله تصمیم گیری برای یک مسئله سخت است، کافی است نشان دهیم که مسئله بهینه سازی معادل آن نیز سخت است.

اگر بخواهیم ثابت کنیم یک مسئله سخت است، ثابت می کنیم مسئله تصمیم گیری آن سخت است.

Reduction

فرض کنید مسائل A و B ، مسائل تصمیم گیری هستند.

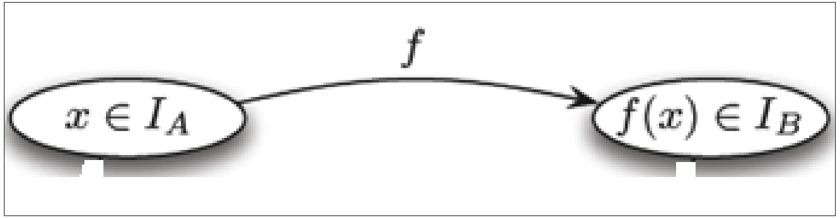
می گوییم A، Reduced شده به B، $A \preccurlyeq_P B$

اگر الگوریتم چند جمله ای f وجود داشته باشد که :

۱- هر نمونه دلخواه از A را به نمونه ای از B تبدیل کند.

۲- جواب ها معادل باشند.(اگر جواب مسئله تصمیم گیری برای نمونه A برابر yes باشد، برای نمونه B هم جواب yes باشد و برعکس.)

- $x \in Instance(A) \Longrightarrow f(x) \in Instance(B)$.
- x is a yes-instance of $A \Longleftrightarrow f(x)$ is a yes-instance of B.



کاربرد Reduction

سختی A از سختی B کمتر است. $A \preccurlyeq_P B$

اگر حل A سخت باشد ، سپس حل B نیز سخت است.

اگر حل B ساده باشد ، سپس حل A نیز ساده است.

اگر حل A ساده باشد ، در رابطه با B نمی توان نظر داد.

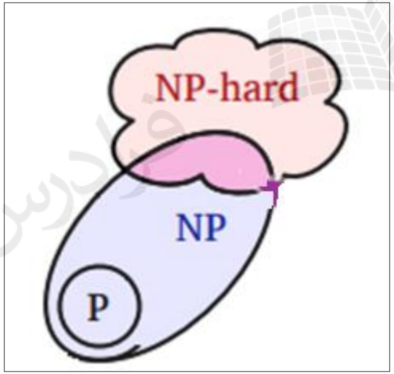
اگر حل B سخت باشد ، در رابطه با A نمی توان نظر داد.

برای اینکه نشان دهیم B سخت است، کافی است یک مسئله شناخت شده سخت به نام A را انتخاب کرده و به B ، Reduction بزنیم.

NP-Hard

همه مسائلی که از کل NP ها سخت تر هستند.
مسئله L ، NP-Hard است اگر همه مسائل NP را بتوان در زمان چند جمله ای به آن redeuce زد.

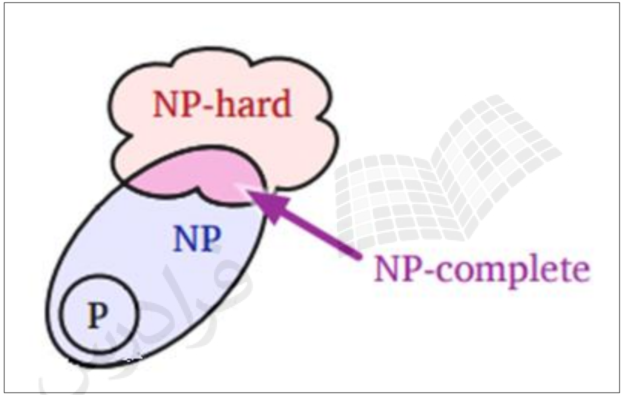
$$L \in NP-Hard \Longleftrightarrow \forall L' \in NP \ : \ L' \preccurlyeq_P L.$$



همه مسائل NP-hard ، NP نمی باشند.
بنابراین پیشوند NP در NP-hard ، معنی non-deterministic polynomial time را نمی دهد.
که این گیج کننده است ولی بعید است که تغییری در این نام گذاری داده شود.

NP-complete

یک مسئله تصمیم گیری NP-complete است،اگر هم NP باشد و هم NP-hard .



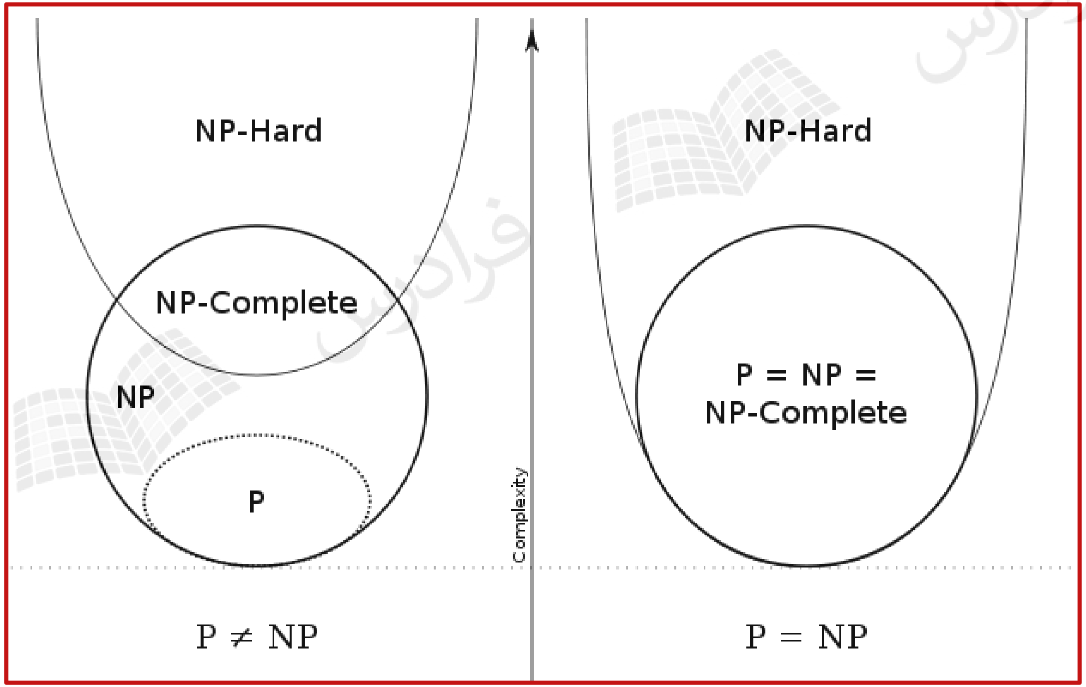
NP-complete problems

A problem X is NP-complete if

- it is in NP
- for every problem $Y \in NP$: $Y \leq_p X$.

Theorem

Let X be an NP-complete problem. If we find a polynomial time algorithm for X, then $P = NP$.



چند مسئله NP- complete

- Boolean satisfiability problem (SAT)
- Clique problem
- Vertex cover problem
- Graph coloring problem
- Knapsack problem
- Subset sum problem
- Subgraph isomorphism problem
- Hamiltonian path problem
- Travelling salesman problem (decision version)

مسئله SAT

Given a Boolean expression on n variables, can we assign values such that the expression is TRUE?

SAT: $(A \vee \neg B \vee \neg C) \wedge (\neg A \vee B) \wedge (\neg B \vee D \vee F) \wedge (F \vee \neg D)$

2SAT: $(\neg x \vee y) \wedge (\neg y \vee z) \wedge (x \vee \neg z) \wedge (z \vee y)$

3SAT: $(a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$

2SAT is a polynomial-time solvable problem.

3SAT is an NP-Complete problem.

آقای کوک در سال ۱۹۷۱ نشان داد که مسئله SAT یک مسئله **NP-Complete** است.

اثبات NP-Complete بودن 3SAT

باید نشان داد که :

۱- **NP** است.

الف- یک الگوریتم **NP** ارائه داد.

یا :

ب- در زمان چند جمله ای **verify** کرد.

$$\boxed{(a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})}$$

۲- **NP-Hard** است.

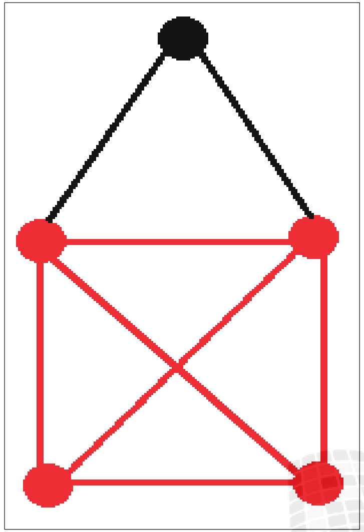
الف- استفاده از تعریف(سخت)

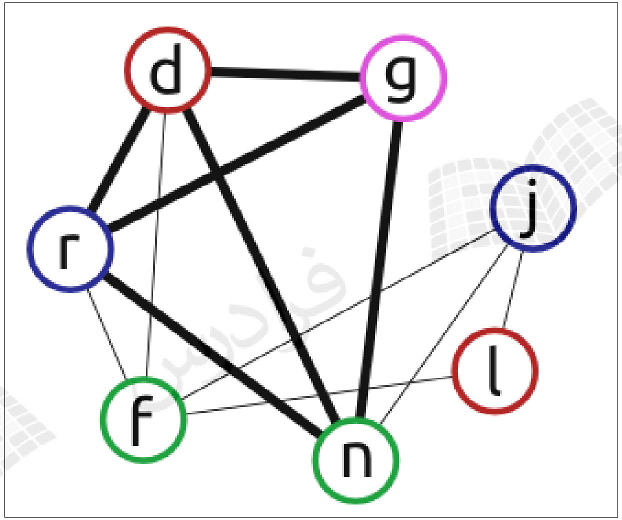
یا:

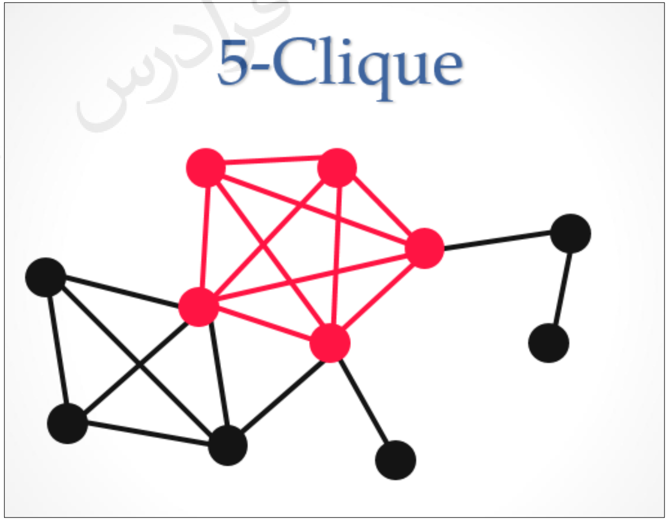
ب- از یک مسئله شناخته شده سخت (مانند **SAT**) به این مسئله **Reduce** زد.

$$SAT \leq_p 3SAT$$

مسئله Clique







اثبات NP-Complete بودن K-Clique

۱- NP است.

چون می توان در زمان چند جمله ای verify کرد.

یعنی بتوان به سادگی این ادعا که k راس گراف داده شده تشکیل کلیک می دهد را بررسی کرد.

۲- NP-Hard است.

چون می توان از یک مسئله شناخته شده سخت ، به این مسئله، Reduce زد.

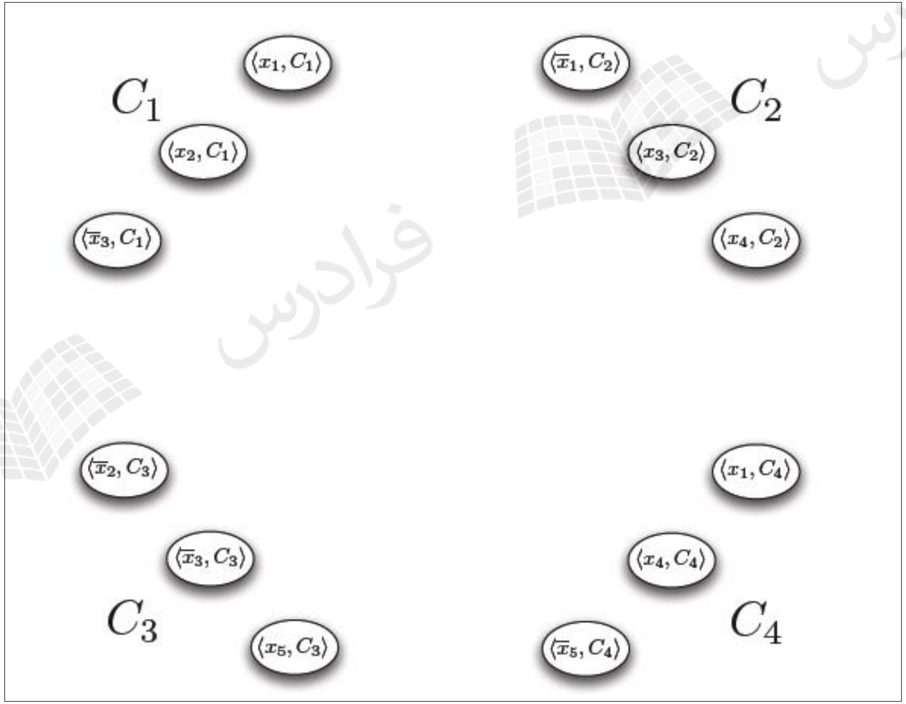
$3SAT \leq_p CLIQUE$

یک فرمول دلخواه در فرم 3SAT را برداشته و یک الگوریتم چندجمله ای نوشته که یک گراف را بدهد.
همچنین جواب ها باید معادل باشد.

مثال (ایجاد گراف از روی عبارت)

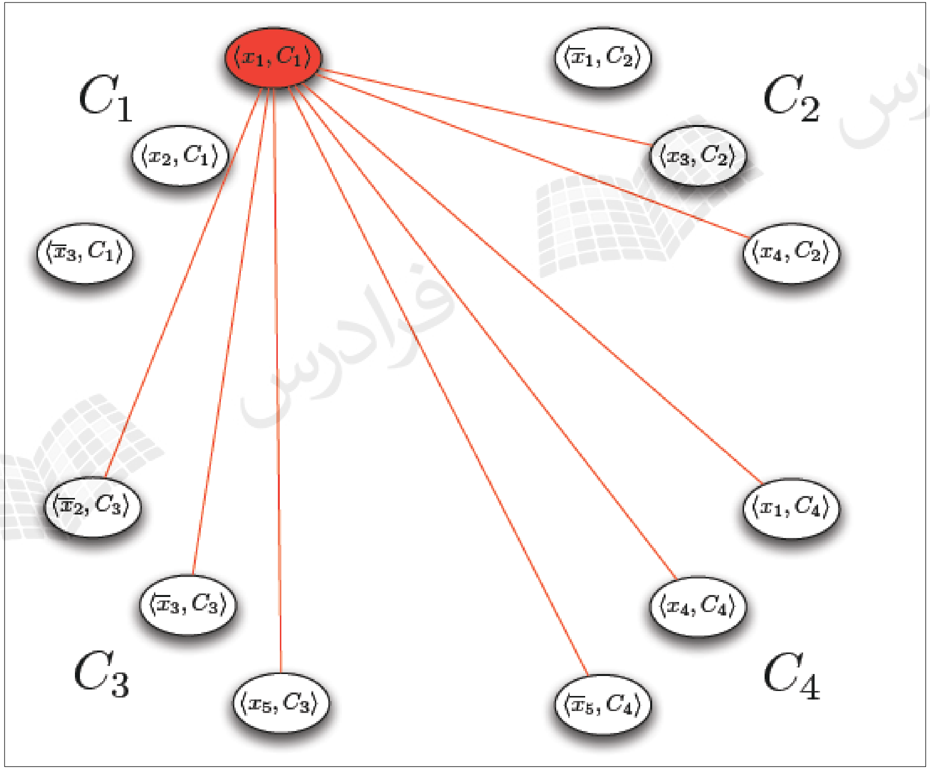
$$\phi = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_3 \vee x_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee x_5) \wedge (x_1 \vee x_4 \vee \overline{x}_5)$$

مرحله اول: ایجاد راس های گراف

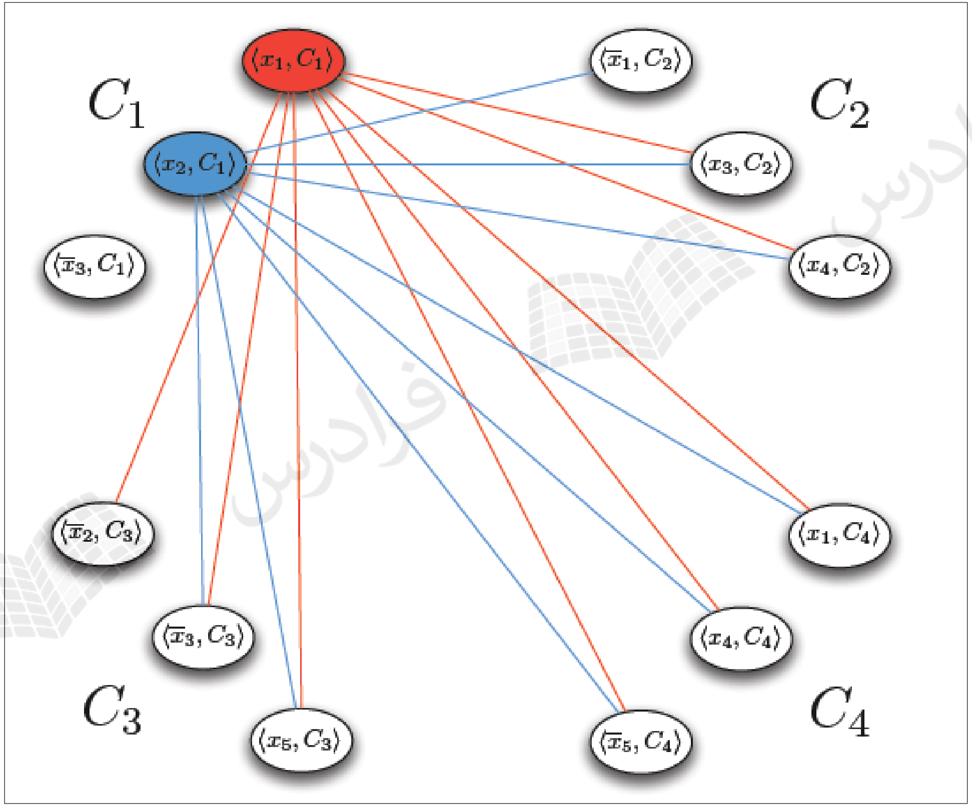


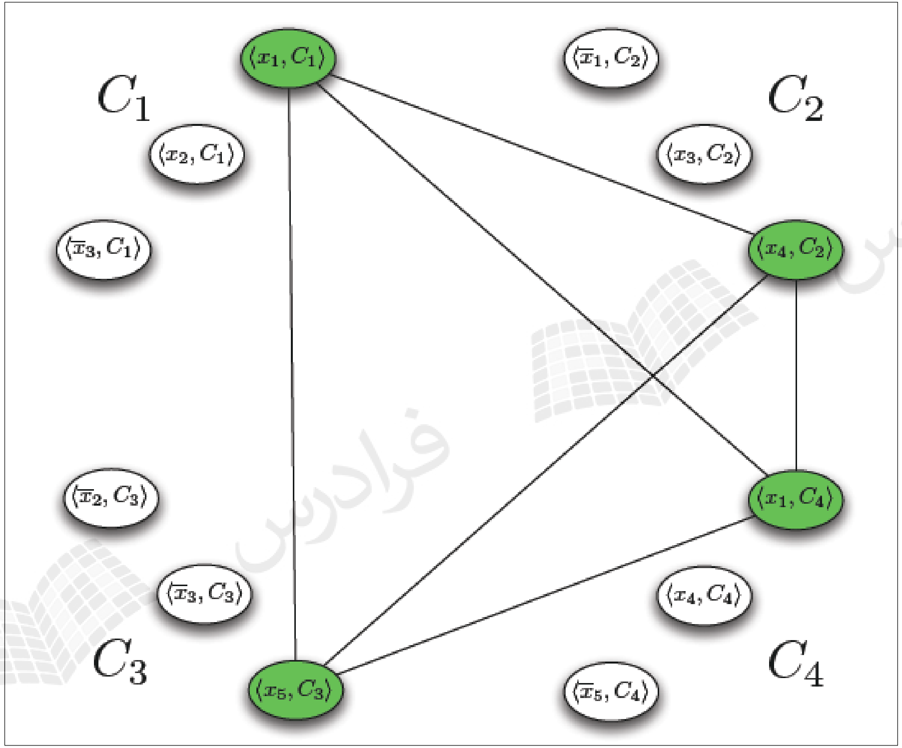
مرحله دوم : ایجاد یال های گراف

۱- در هر کلاس ، هیچ راسی به راسی دیگر وصل نیست.
۲- هر متغیر به نقیض خود در کلاس دیگر وصل نیست.



ادامه ایجاد یال ها





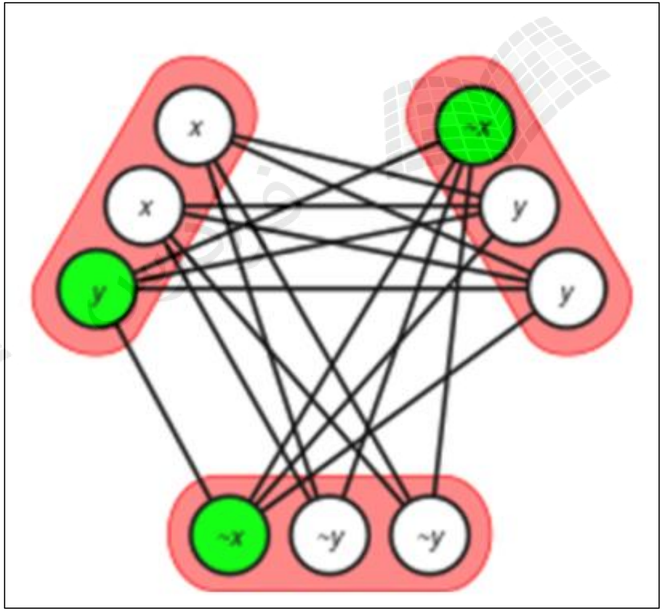
اگر در C1 مقدار X1 ، در C2 مقدار X4 ، در C3 مقدار X5 و در C4 مقدار X1 برابر یک باشد، آنگاه فرمول TRUE شده و در گراف هم یک کلیک خواهیم داشت.

Reduce یک نمونه 3SAT به مسئله کلیک

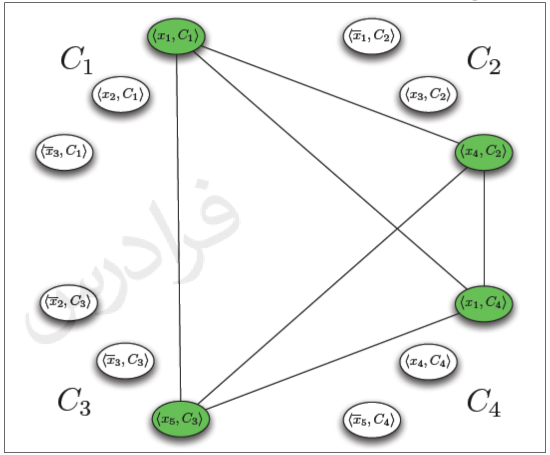
$$(x \vee x \vee y) \wedge (\neg x \vee \neg y \vee \neg y) \wedge (\neg x \vee y \vee y)$$

$$x = 0$$

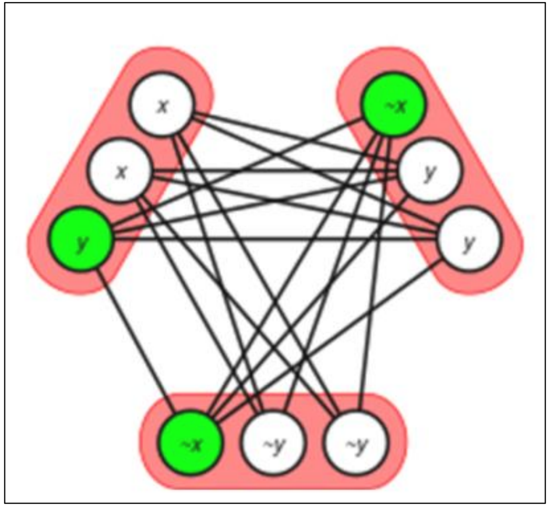
$$y = 1$$



$$\phi = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_3 \vee x_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee x_5) \wedge (x_1 \vee x_4 \vee \overline{x}_5)$$

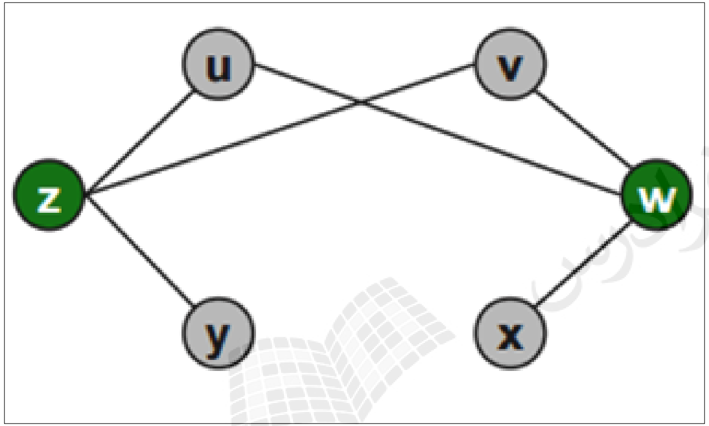


$$(x \vee x \vee y) \wedge (\neg x \vee \neg y \vee \neg y) \wedge (\neg x \vee y \vee y)$$

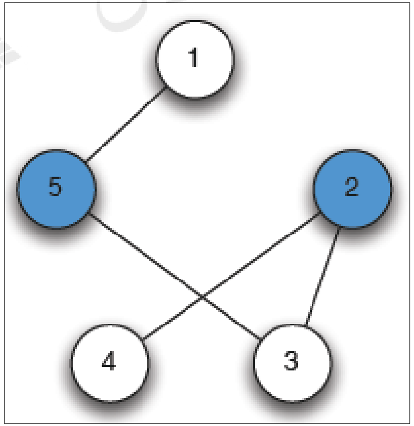


مسئله Vertex Cover

پیدا کردن زیر مجموعه ای از گره های گراف که همه یال ها را بپوشاند.
یعنی حداقل یک راس هر یال گراف، در این مجموعه باشد.



Vertex cover={w,z}



Vertex cover={2,5}

K-Vertex Cover

Suppose that a graph $G = (V, E)$ is given.

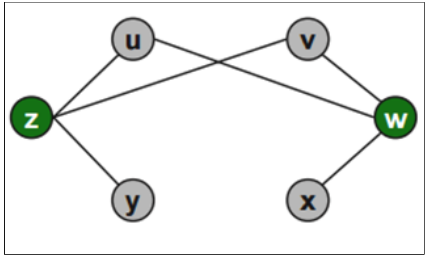
- Optimization: Find a minimal subset $V' \subseteq V$, such that for each $e = (v_1, v_2) \in E$, either $v_1 \in V'$ or $v_2 \in V'$ (V' covers the E).
- Decision: For a given k, is there a subset $V' \subseteq V$ with $|V'| \leq k$, such that V' covers the E?

اثبات NP-Complete بودن K-Vertex Cover

۱– NP است.

چون می توان در زمان چند جمله ای verify کرد.

اگر یک زیر مجموعه از راس ها به ما بدهند وبخواهیم چک کنیم که آیا VC است یا نه، کافی است یکی یکی یال ها را نگاه کرده و اگر حداقل یکی از راس های گراف در این مجموعه بود، جواب درست خواهد بود.
حداکثر به اندازه یالهای گراف عمل مقایسه انجام می شود پس چند جمله ای است.

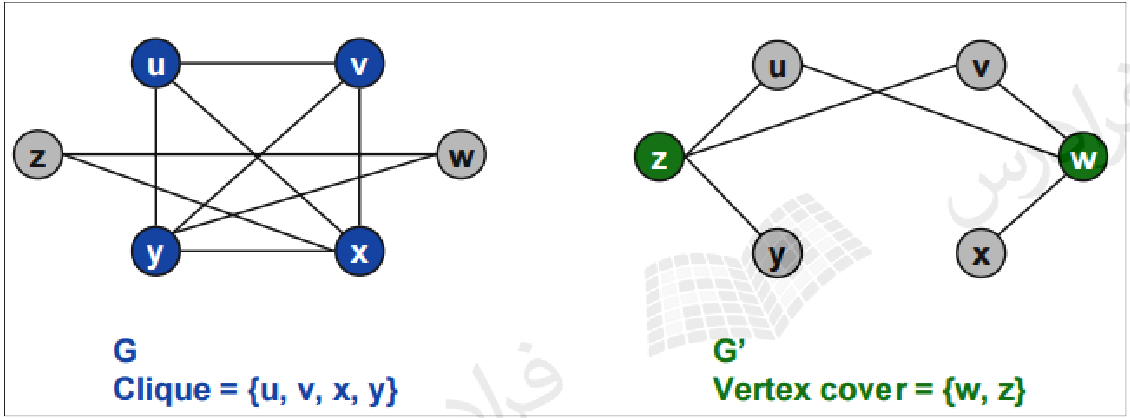


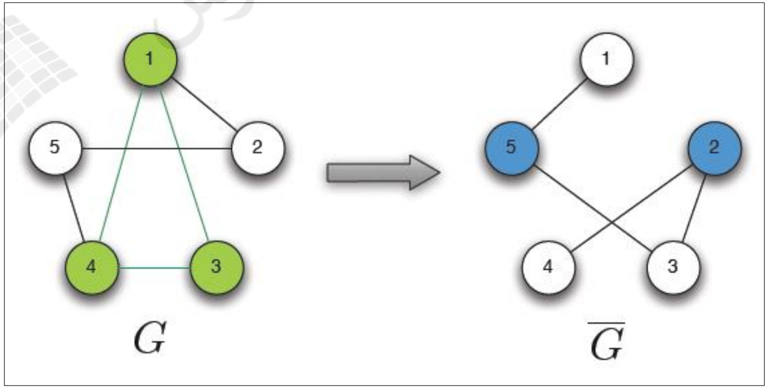
Vertex cover={w,z}

۲– NP-Hard است.

چون می توان از یک مسئله شناخته شده سخت مانند K-Clique ، به این مسئله، Reduce زد.

یک گراف دارای یک کلیک به اندازه k است ، اگر و فقط اگر مکمل گراف دارای یک vertex cover به اندازه |V| - k باشد.





Subset Sum Problem

Suppose that a set $S = \{x_1, x_2, \cdots, x_n\}$ and a positive integer t are given. Is there a subset $S' \subseteq S$ such that $\sum_{x \in S'} x = t$?

S ={1,4,16,64,256,1040,1041,1093,1284,1344}

t = 3754

s'={1,16,64,256,1040,1093,1284}

اثبات NP-Complete بودن مسئله Subset Sum

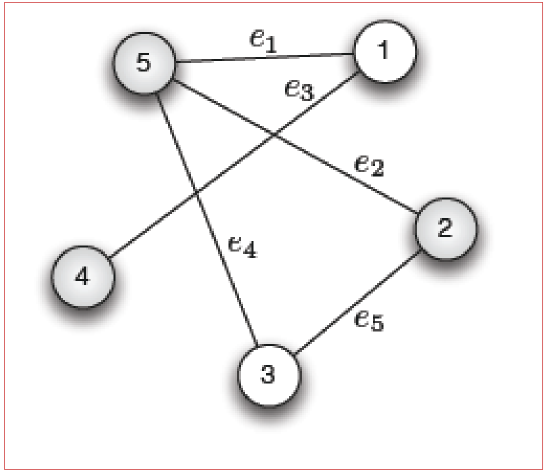
۱– NP است.

چون می توان در زمان چند جمله ای verify کرد.

s'={1,16,64,256,1040,1093,1284}
t = 3754
S ={1,4,16,64,256,1040,1041,1093,1284,1344}

۲– NP-Hard است.

$$\text{VERTEX-COVER} \le_P \text{SUBSET-SUM}.$$



M=

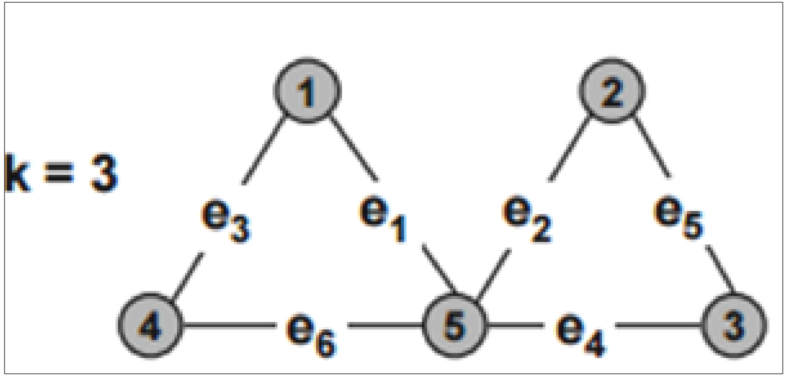
	e_1	e_2	e_3	e_4	e_5
v_1	1	0	1	0	0
v_2	0	1	0	0	1
v_3	0	0	0	1	1
v_4	0	0	1	0	0
v_5	1	1	0	1	0

	MSB	e_1	e_2	e_3	e_4	e_5	
x_1	1	1	0	1	0	0	1296
x_2	1	0	1	0	0	1	1089
x_3	1	0	0	0	1	1	1029
x_4	1	0	0	1	0	0	1040
x_5	1	1	1	0	1	0	1348
y_1	0	1	0	0	0	0	256
y_2	0	0	1	0	0	0	64
y_3	0	0	0	1	0	0	16
y_4	0	0	0	0	1	0	4
y_5	0	0	0	0	0	1	1
t	k	2	2	2	2	2	3754

$$0\times 4^0+0\times 4^1+1\times 4^2$$
$$+0\times 4^3+1\times 4^4+1\times 4^5$$
$$=1296$$

1089+1040+1348+256+16+4+1=3754

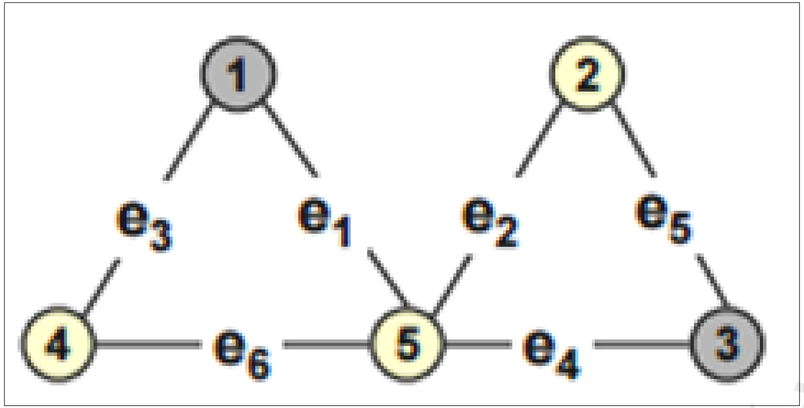
Subset Sum



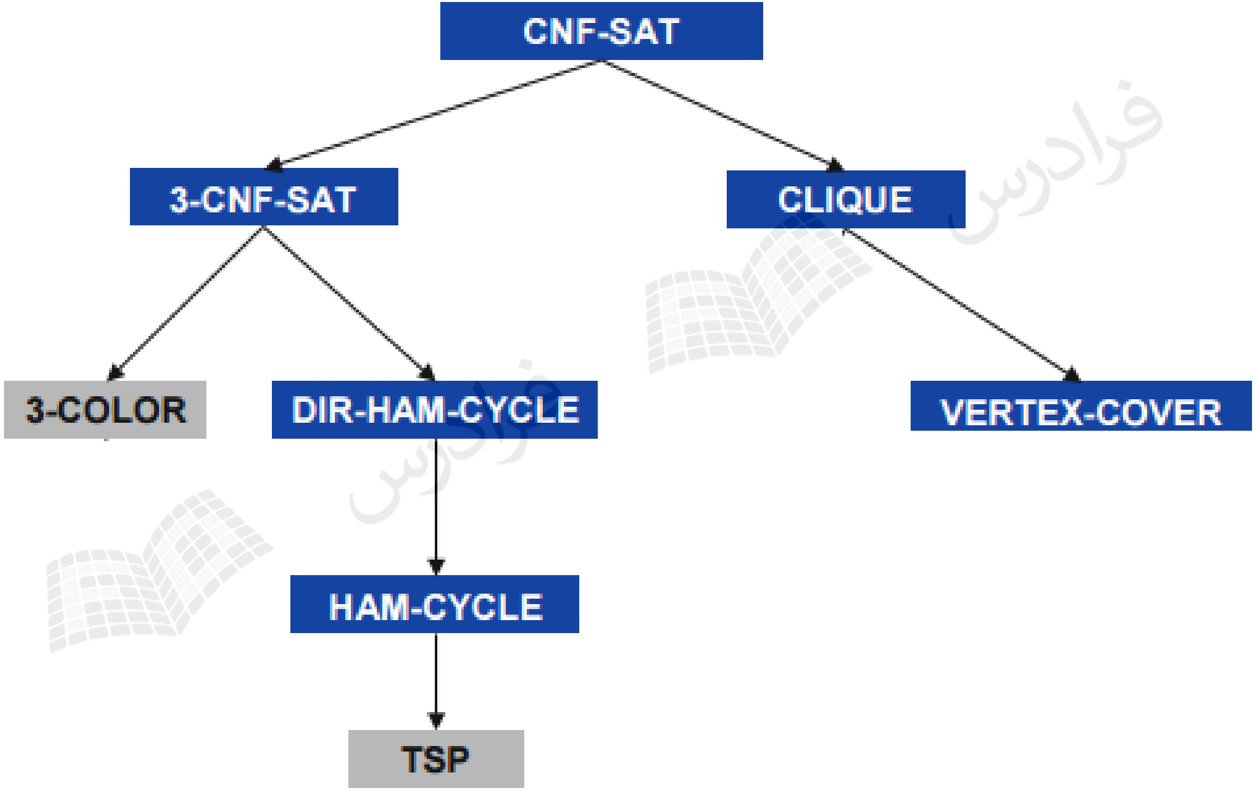
	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1	0	1	0	0	0
v_2	0	1	0	0	1	0
v_3	0	0	0	1	1	0
v_4	0	0	1	0	0	1
v_5	1	1	0	1	0	1

incidence matrix

	k	e_1	e_2	e_3	e_4	e_5	e_6	decimal
x_1	1	1	0	1	0	0	0	5,184
x_2	1	0	1	0	0	1	0	4,356
x_3	1	0	0	0	1	1	0	4,116
x_4	1	0	0	1	0	0	1	4,161
x_5	1	1	1	0	1	0	1	5,393
y_1	0	1	0	0	0	0	0	1,024
y_2	0	0	1	0	0	0	0	256
y_3	0	0	0	1	0	0	0	64
y_4	0	0	0	0	1	0	0	16
y_5	0	0	0	0	0	1	0	4
y_6	0	0	0	0	0	0	1	1
t	3	2	2	2	2	2	2	15,018



		e_1	e_2	e_3	e_4	e_5	e_6	decimal
x_1	1	1	0	1	0	0	0	5,184
x_2	1	0	1	0	0	1	0	4,356
x_3	1	0	0	0	1	1	0	4,116
x_4	1	0	0	1	0	0	1	4,161
x_5	1	1	1	0	1	0	1	5,393
y_1	0	1	0	0	0	0	0	1,024
y_2	0	0	1	0	0	0	0	256
y_3	0	0	0	1	0	0	0	64
y_4	0	0	0	0	1	0	0	16
y_5	0	0	0	0	0	1	0	4
y_6	0	0	0	0	0	0	1	1
t	3	2	2	2	2	2	2	15,018
	k							


CNF-SAT
3-CNF-SAT
CLIQUE
3-COLOR
DIR-HAM-CYCLE
VERTEX-COVER
HAM-CYCLE
TSP

پایان

مشاوره با مدرس شیرافکن

۰۹۱۲۱۹۷۲۰۲۸

این اسلاید ها بر مبنای نکات مطرح شده در فرادرس

«آموزش طراحی الگوریتم»

تهیه شده است.

برای کسب اطلاعات بیشتر در مورد این آموزش به لینک زیر مراجعه نمایید

faradars.org/fvsft1092