BARVANJE GRAFOV IN DINAMIčNO PROGRAMIRANJE

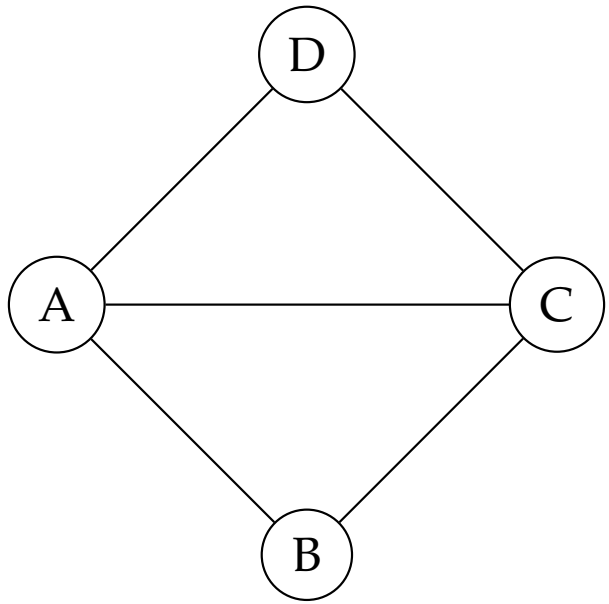
Barvanje grafov

- Dodelitev "barv" vozliščem v grafu, krajišči povezave sta v drugih barvah.

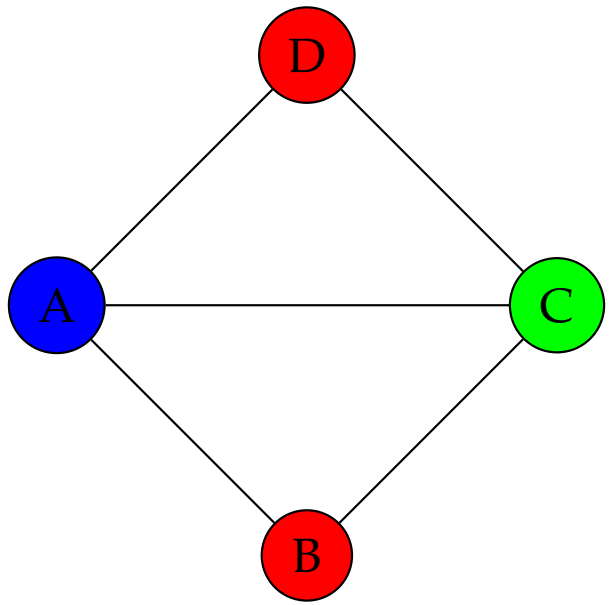
$$col : V \to C, \forall (u, v) \in E, col(u) \neq col(v)$$

- Tipično je to minimizacijski problem, kjer minimiziramo število uporabljenih barv ($|C|$)
- Najmanjše možno število barv v grafu je kromatsko število grafa $\chi(G)$.
- Realne aplikacije: problemi v logistiki, dodeljevanje registrov v prevajalnikih, priprava urnikov, reševanje drugih težkih kombinatoričnih problemov.

Primer



Obarvan primer



Algoritmična zahtevnost

- Problem je NP-težek
- Velikost preiskovalnega prostora je (naivno) k^n - vsakemu vozlišču lahko dodelimo eno od k možnih barv (v najslabšem primeru je to n).
- obstajajo boljši natančni algoritmi - najboljši temelji na dinamičnem programiranju $O(2.4423^n)$

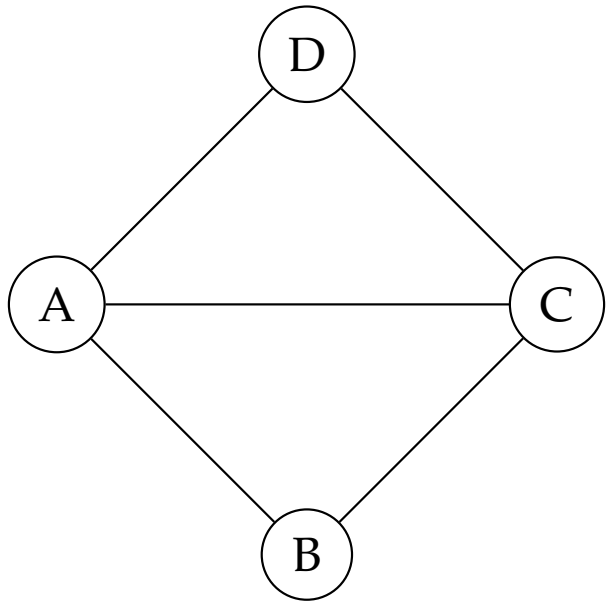
Povezava z neodvisnimi množicami

1. Neodvisna množica:
   - Neodvisna množica v grafu $G = (V, E)$ je (pod)množica vozlišč $S \subseteq V$ za katero velja $\forall i, j \in S : (i, j) \notin E$. Po domače - med nobenim parom vozlišč v tej množici ni nobene povezave.
   - Vozlišča v neki neodvisni množici so lahko obarvana z isto barvo
2. Rekurzivna relacija za $\chi(G)$:

$$\chi(G) = 1 + min(\chi(G[V \setminus I])),$$

kjer je minimum vzet preko vseh neodvisnih množic grafa G, $G[V \setminus I]$ pa predstavlja graf iz katerega je odstranjena množica vozlišč I .

Neodvisne množice



Neodvisna (netrivialna) množica tega grafa je ena sama : $\{D, B\}$. Če iz grafa odstranimo to množico, dobimo eno samo povezavo, ki jo je mogoče obarvati z dvema barvama.

$$\chi(G) = 1 + \chi([G \setminus \{D, B\}]) = 3$$

Implementacijske podrobnosti

- Rekurzivni izračun po formuli je praktično nemogoč
- S hranjenjem manjših rešitev ostane izvajanje bistveno bolj obvladljivo (še vedno seveda eksponentno)
- Za vse $S \subseteq V$ naračunamo $\chi(S)$

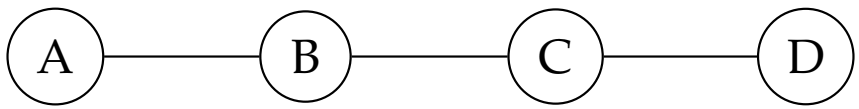
Bitna predstavitev podmnožic

- Rešitve predstavimo v tabeli velikosti 2^n
- Indeks v tej tabeli predstavlja podmnožico, npr. pri $V = \{A, B, C, D\}$ index $i = 6$ (binarno 0110) predstavlja podmnožico $\{B, C\}$.
- Za vse $S \subseteq V$ naračunamo $\chi(S)$
- Če pomnožice uredimo po velikosti (začnemo pri najmanjši), potem so manjše rešitve že na voljo, ko jih potrebujemo.

Generiranje neodvisnih podmnožic

- Uporabimo naiven sprehod po vseh podmnožicah neke množice in ugotavljamo katere so neodvisne
- Obstajajo sicer bolj učinkovite metode - izven domene te predstavitve

Primer



$\chi(\emptyset) = 0$	$\chi(\{A\}) = 1$	$\chi(\{B\}) = 1$	$\chi(\{C\}) = 1$
$\chi(\{D\}) = 1$	$\chi(\{A, B\}) = 2$	$\chi(\{A, C\}) = 1$	$\chi(\{A, D\}) = 1$
$\chi(\{B, C\}) = 2$	$\chi(\{B, D\}) = 1$	$\chi(\{C, D\}) = 2$	$\chi(\{A, B, C\}) = 2$
$\chi(\{A, B, D\}) = 2$	$\chi(\{A, C, D\}) = 2$	$\chi(\{B, C, D\}) = 2$	$\chi(\{A, B, C, D\}) = 2$