

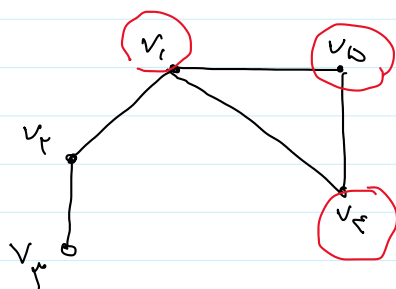
موضوع: گراف - مقدمات. DFS و BFS

گراف (کوتاهترین مسیر - درخت پوشای مین - تطابق در گراف دو بخشی)
شمار بستینه

هر گراف به صورت یک دوتایی $\langle V, E \rangle$ است که در آن:

۱- V مجموعه رئوس گراف است $V = \{v_1, v_2, \dots, v_n\}$ - تعداد رئوسها n :

۲- E مجموعه یالهای گراف است. هر یال یک دوتایی (v_i, v_j) است که $v_i, v_j \in V$:



$$V = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E = \{(v_1, v_2), (v_1, v_5), (v_1, v_4), (v_2, v_3), (v_5, v_4)\}$$

تعاریف ابتدایی:

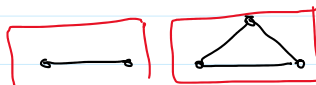
همسایه: $N(v_i) = \{v_1, v_4\}$ $N(v_5)$

درج: d_i : درج راس v_i

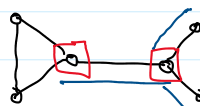
مسیر:

دور:

گراف همبند:



گراف ناهمبند:



راس برشی:

یال برشی:

گراف کامل:

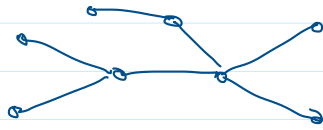
...

رُف = $\frac{m}{n}$:

زیر گراف :

زیر گراف متبانی :

انواع گراف ها :

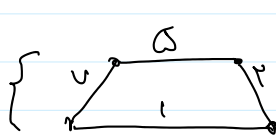


درخت : گراف همبند بدون دور :

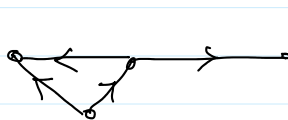
- ۱- تعداد یال های درخت : $n-1$
- ۲- همه یال های درخت برشی
- ۳- همه راس های غیر برگ ، برشی هستند . ← راس با درجه ۱

* ۴- اضافه کردن هر یال به درخت ایجاد دوری کند .

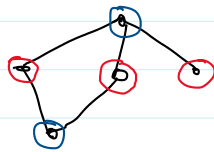
تعداد درخت ها با n راس v_1, v_2, \dots, v_n : n^{n-2} *



گراف های وزن دار : گرافی که وزن دارد .



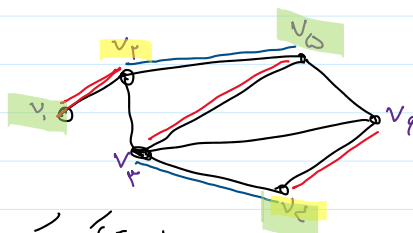
گراف های جهت دار : گرافی که جهت دارد .



گراف k بخشی (بخشی) :

تطابق Matching : یک زیر مجموعه از یال های گراف تشکیل یک

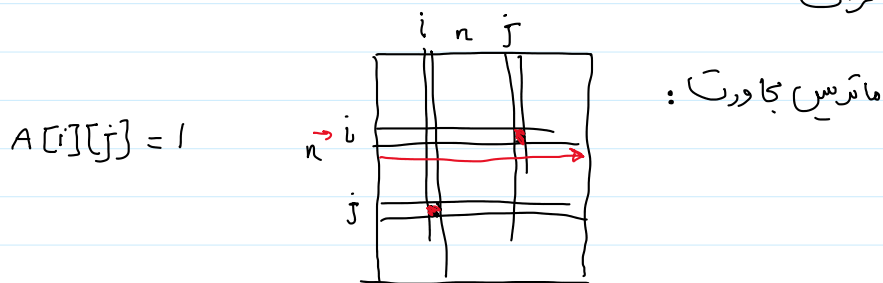
تطابق می دهند اگر هیچ دو یالی راس مشترک نداشته باشند



مجموعه مستقل independent set : یک زیر مجموعه از راس ها ، تشکیل یک مجموعه مستقل

مجموعه مستقل independent set: یک زیرمجموعه از رئوس ها، تشکیل یک مجموعه مستقل می دهند، اگر بین هیچ دو رئوس از مجموعه یالی نباشد.

* پیاده سازی گراف

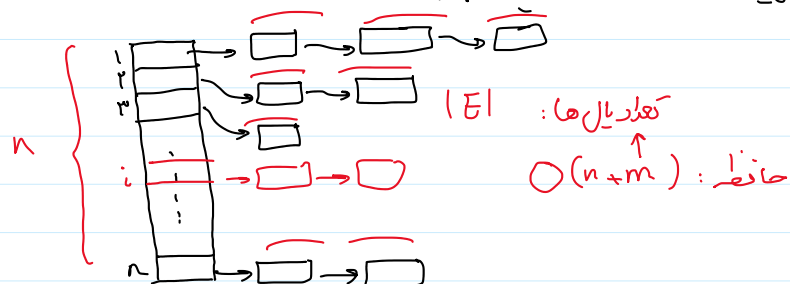


حافظه: $O(n^2)$

query1: آیا رئوس v_i و v_j همسایه هستند؟ $O(1)$

query2: ارائه لیستی از همسایه های v_i (رئوس): $O(n)$

پیاده سازی 2: لیست مجاورت:



* query1: آیا بین v_i و v_j یال است؟ $O(d_i)$

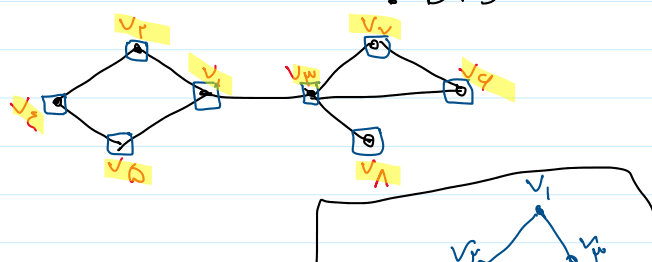
* query2: ارائه لیستی از همسایه های v_i : $O(d_i)$

$$* O(m) = \sum_i d_i = 2m$$

پیمایش گراف: BFS - DFS
اول برآورد (رئوس): mark

: DFS

DFS(v_i) {
mark[v_i] = true
for every $v_j \in N(v_i)$
if (!mark[v_j])

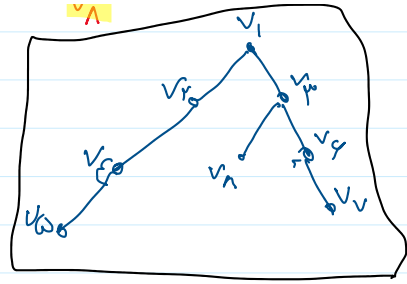


if (!mark[vj])
DFS(vj)

}

DFS(v1):

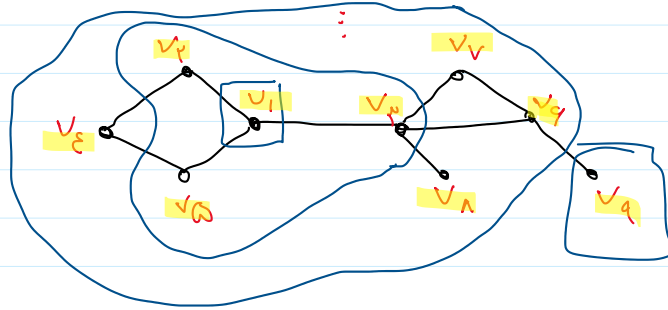
DFS درخت ←



کاربرد: ۱- پیدا کردن مولفه های همبندی ۲- آیا بین v_i و v_j مسیری هست؟
۳- پیدا کردن راس ها و یال های برشی ۴- پیدا کردن مولفه های قویا

زمان اجرا: $O(n^2)$ ماتریس مجاورت
لیست مجاورت $O(n+m)$ \leftarrow $\sum d_i = 2m$

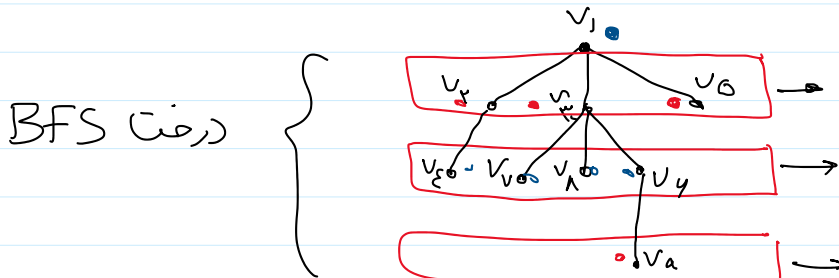
BFS : BFS(v1) : ۱- خود راس v_i
۲- همسایه های راس v_i
۳- همسایه های همسایه های v_i



BFS(v1):

Q:

v1	v2	v3	v4	v5	v6	v7	v8	v9
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------



کاربرد: * کوتاه ترین مسیر \sum زمان: $O(n^2)$ ماتریس مجاورت

* تست دو بخشی بودن گراف.

لیست مجاورت: $O(n+m)$