RECAP: Christofides algorithm: determine an Eulerian multi-subgraph $\bar{G}$ of (G, c) spanning V of small cost. Obtain Euler Tour touching all edges once and from it a 'tour' via shortcutting:

CHRISTOFIDES $(G=(V,E), c)$ — $O(|E|\log|V|) = O(|V|^2 \log|V|)$

$T^* = (V, E_{T^*}) \leftarrow$ MST(G, c)

* let $V^{T^*}_{ODD}$ be the set of nodes of odd degree of T^* and let $E^{T^*}_{ODD} = \{e = \{u,v\} \in E : u, v \in V^{T^*}_{ODD}\}$ * — $O(|V|)$

$M^* \leftarrow$ M_C_P_MATCHING$(G' = (V^{T^*}_{ODD}, E^{T^*}_{ODD}), c)$ — $O(|V|^3)$

$W \leftarrow$ EULER_TOUR$(\bar{G} = (V, E_{T^*} \cup M^*))$ — $O(|V|)$

$\gamma \leftarrow$ SHORTCUT(W) — $O(|V|)$

return γ

Running Time $O(|V^3|)$ (Edmond's Blossom Algorithm for min-cost perfect matching)

Approximation ratio: Since $c(M^*) \leq c(\gamma^*)/2$

$$c(\gamma) \leq c(E_{T^*}) + c(M^*) \leq \frac{3}{2} c(\gamma^*)$$

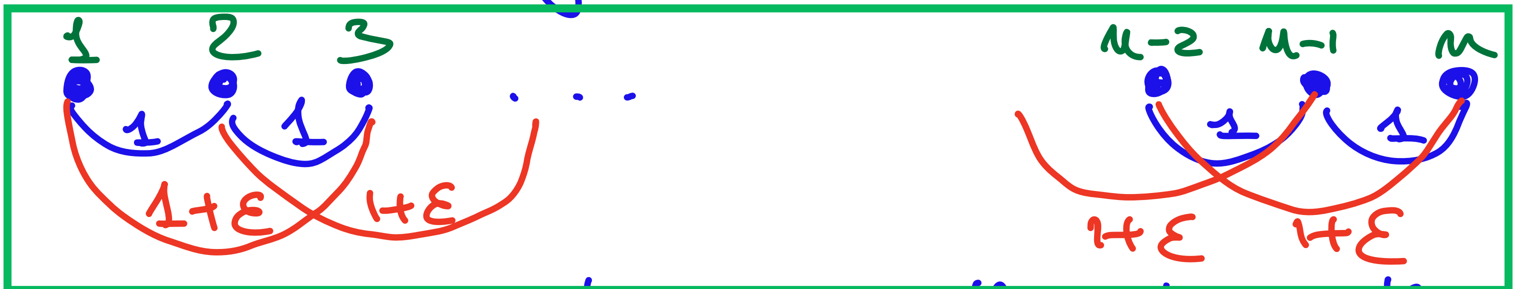
$$\Rightarrow \rho = 3/2$$

It can be proved that the bound on $\rho(|N|)$ is tight, in the sense that there is a family of weighted complete graphs $\boxed{G_n = (V_n, E_n), c_n \quad (|V_n| = n)}$

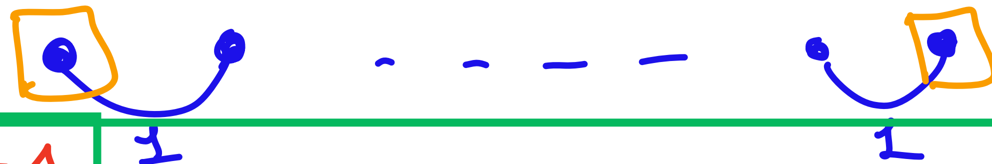
$$\boxed{\lim_{n \to +\infty} \rho_{G_n}(n) = \frac{3}{2}}$$

The graph: let n be odd:

For $\varepsilon > 0$ very small (to be determined):



To complete the graph: weigh all other edges with cost of the shortest path $\pi^*_{u,v}$

MST T^*: 

$\boxed{C(T^*) = n-1}$

MCPM of $V_{ODD} = \boxed{\{1, n\}}$

$\boxed{M^*}$: one edge $\{1, n\}$ of cost

$$\boxed{\left\lfloor \frac{n}{2} \right\rfloor (1+\varepsilon) \quad (n \text{ odd})}$$

$\boxed{T^* \cup M^*}$ is the tour γ (no shortcutting). $\boxed{C(T^* \cup M^*) \simeq \frac{3}{2} n + O(\varepsilon \cdot n)}$

$$\gamma^* = (1, 3, \dots, n, n-1, n-3, \dots, 2, 1)$$

$$C(\gamma^*) = \left\lfloor \frac{n}{2} \right\rfloor (1+\varepsilon) + 2 + \left(\left\lfloor \frac{n}{2} \right\rfloor - 1 \right)(1+\varepsilon)$$

$$= n + O(\varepsilon n)$$

Choose $\varepsilon = \frac{1}{n^2}$: $\lim_{n \to +\infty} \frac{C(T^* \cup M^*)}{C(\gamma^*)} = \frac{3}{2}$

INAPPROXIMABILITY RESULT:

For TRIANGLE-TSP there cannot exist a constant ρ-approximation algorithm for

$$\boxed{\rho < \frac{123}{122} \simeq 1{,}008}$$

unless P = NP [Karpinski, Lampis, Schmied '15]

Christofides' algorithm has been the best approximation algorithm for Δ-TSP for 48 years (since 1976).

In 2020, a new (impractical) algorithm has been announced with

$$\rho \le \frac{3}{2} - 10^{-36}$$

[Karlin, Klein, Gharan '20]

FURTHER RESTRICTION: EUCLIDEAN TSP

The nodes of G are points $\in \mathbb{R}^d$ and $\boxed{c(u,v) = \|u-v\|_2 = d_E(u,v)}$

There is a PTAS for EUCLIDEAN TSP

Running time: $\boxed{O\left(n\left(\log n + 2^{\mathrm{poly}(1/\varepsilon)}\right)\right)}$

The algorithm is totally impractical

THE (MINIMUM) SET COVER (SC) PROBLEM

An instance of SC is a pair $(X, \mathcal{F})$

X : universe of elements

$\mathcal{F}$: family of subsets of X : $\mathcal{F} \subseteq \{S : S \subseteq X\}$

$[S \in \mathcal{F} \Rightarrow S \subseteq X]$

$\mathcal{P}(X)$ boolean of X

Covering constraint: $X = \bigcup_{S \in \mathcal{F}} S$

$(\forall x \in X \ \exists S \in \mathcal{F} : x \in S)$

A subset $\mathcal{C} \subseteq \mathcal{F}$ is a covering of X (also: $\mathcal{C}$ covers X) if $X = \bigcup_{S \in \mathcal{C}} S$

(observe that $\mathcal{F}$ covers X)

We wish to find a minimum cardinality covering $\mathcal{C}^*$ of X

SC is an important problem which extends Vertex Cover

APPLICATIONS:

X = set of individuals

$\mathcal{F}$ = set of mailing lists

$|\mathcal{C}^*|$: minimum number of bulk-e-mails to reach all individuals

X' = set of required skills

$\mathcal{F}'$ = set of individuals (each represented by the set of his/her skills)

$|\mathcal{C}^*|$: minimum number of hirings to cover all skills.

The decision version is the following

SC

I: $\langle X, \mathcal{F}, K \rangle$

Q: $\exists \mathcal{C} \subseteq \mathcal{F}$ covering X, with $|\mathcal{C}| \leq K$?

We prove that $\boxed{SC \in NPH}$ via $\boxed{VC \leq_p SC}$

The reduction simply "reformulates" VC as a set-cover problem (generalization)

Given $\boxed{\langle G=(V,E), K \rangle}$:

$$\boxed{f(\langle G=(V,E), K \rangle) = \langle X_G, \mathcal{F}_G, K' \rangle}$$

$\boxed{X_G = E}$ (we have to cover edges)

$\boxed{\mathcal{F}_G = \{ N_v : v \in V \}}$

$\boxed{N_v}$: all edges that v can cover:

$$\boxed{N_v = \{ e \in E : e \ni v \}}$$

Clearly:

$\langle G=(V,E), k\rangle \in VC$

$\Leftrightarrow \exists V' \subseteq V, |V'|=k :$
$\forall e=\{u,v\} \in E : (u \in V') \vee (v \in V')$

$\Leftrightarrow \exists V' \subseteq V, |V'|=k :$
$\forall e \in E \; \exists u \in V' : e \in N_u$

$\Leftrightarrow \exists V' \subseteq V, |V'|=k :$
$E = \bigcup_{u \in V'} N_u$

$\Leftrightarrow \exists \mathcal{C}' \subseteq \mathcal{F}_G, |\mathcal{C}'| = k' = k$ $\quad (\mathcal{C}' = \{N_u : u \in V'\})$
$X_G = E = \bigcup_{S \in \mathcal{C}'} S$

$\Leftrightarrow f(\langle G=(V,E), k\rangle) = \langle E_G, \mathcal{F}_G, k'\rangle \in SC$

SC admits the following, intuitive greedy algorithm:

- GC: Choose the subset $S \in \mathcal{F}$ covering the largest number of elements of X
- Cleanup: Remove covered elements

(REMARK: For VC this corresponds to select the node of largest degree)

Pseudocode

```
APPROX-SET-COVER(X, F)
  U ← X ; C ← ∅
  while (U ≠ ∅) do
    S ← argmax{|T ∩ U| : T ∈ F}
    U ← U − S ; C ← C ∪ {S}
  return C
```

NOTE After S is selected, in all subsequent iterations $S \cap U = \emptyset$ ($\Rightarrow$ S will not be selected again)

CORRECTNESS

Since $X = \bigcup_{S \in F} S$, $|U|$ is strictly decreasing hence the algorithm terminates. On termination,

$$\boxed{U = \emptyset \Rightarrow \forall x \in X \; \exists S \in C : x \in S}$$

Thus, $\boxed{C \text{ covers } X}$

RUNNING TIME Trivial Analysis:

- $O(\min(|X|, |F|))$ iterations
- Computing $\mathrm{argmax}\{|T \cap U| : T \in F\}$ can be done in time $O(|X||F|)$

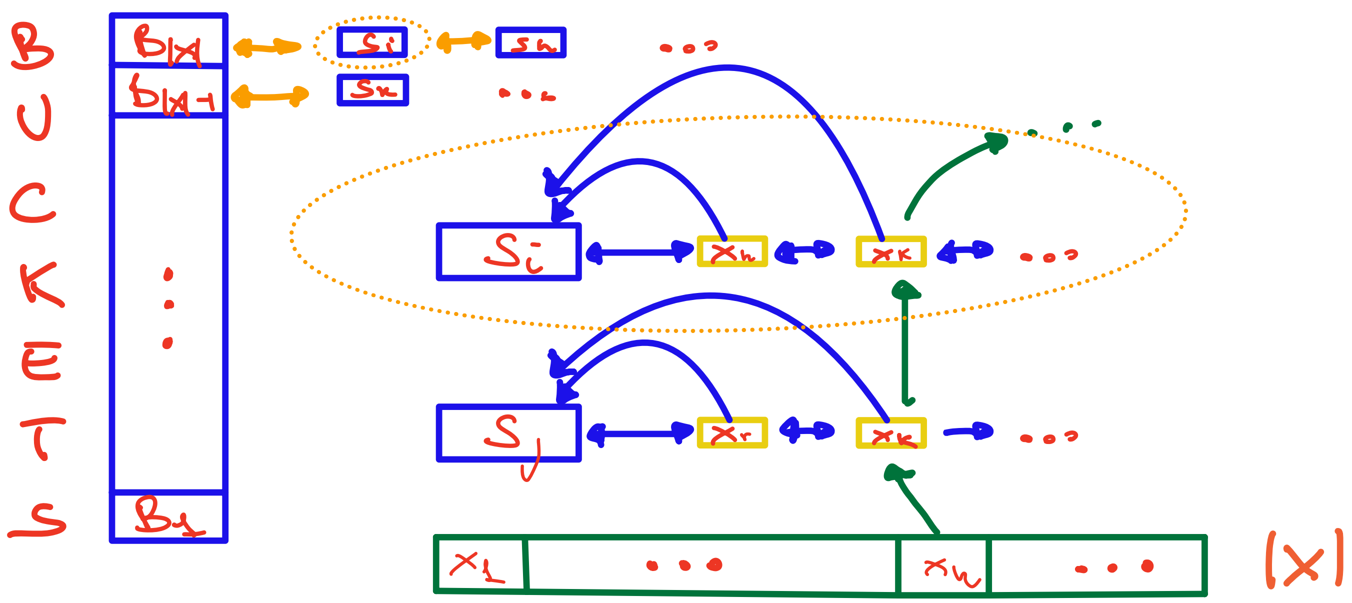
$\rightarrow$ $\boxed{T(|X|, |F|) = O(|X||F| \min\{|X|, |F|\})}$

can be cubic in $|\langle X, F \rangle|$!

A-S-C admits a linked-list-based implementation running in time

$$\boxed{O\left(\sum_{S\in\mathcal{F}}|S|\right) = O(|\langle X,\mathcal{F}\rangle|)}$$

EXERCISE (IDEA: Store each subset as a doubly-linked list of its elements. Also, for each element $x \in X$, keep another list linking all set entries equal to x. Subsets have to be maintained sorted by their cardinality, using $|X|$ list buckets.)



After each selection, the elements' lists can be used to update the subsets and their reference buckets.

We will see two different arguments to upperbound ρ:

- Simple argument yielding $\boxed{\rho(n) \le \lceil \ln n \rceil \text{ with } n = |X|}$

- Complex argument yielding $\boxed{\rho(\langle X, \mathcal{F} \rangle) \le \ln |S_{max}| + 1}$ with $\boxed{|S_{max}| = \max\{|S| : S \in \mathcal{F}\}}$

(NOTE: $|S_{max}|$ can be $\ll n$!)

Interesting charging argument (uses weights)

THEOREM $\boxed{\rho(n) \le \lceil \ln n \rceil, \quad n = |X|}$

```
APPROX_SET_COVER(X, F)
 U ← X; C ← ∅
 while (U ≠ ∅) do
   S ← argmax{|T ∩ U| : T ∈ F}
   U ← U − S; C ← C ∪ {S}
 return C
```

PROOF Call $\boxed{U_t}$ set U at the start of iteration $t \ge 1$. U_t is the set of elements that still need to be covered.

Observe that the first time $\bar{t}$ when $|U_{\bar{t}}| = \emptyset$ implies that $\boxed{|\mathcal{C}| = \bar{t} - 1}$.

Also, call $\boxed{S_t}$ the set selected during iteration t ($\boxed{S_t = \arg\max \{ |T \cap U_t| : T \in \mathcal{F} \}}$)

Let $\boxed{k = |\mathcal{C}^*|}$ (cost of optimal cover)

Since $U_t \subseteq X$ and X admits a cover of size k, $U_t \subseteq X$ can also be covered with ($\leq$) k subsets. Thus:

$$\boxed{\exists \; S_{i_1}, S_{i_2}, \dots, S_{i_k} \in \mathcal{F} : \quad U_t \subseteq \bigcup_{j=1}^{k} S_{i_j}}$$

Then, we can apply the pigeon-hole principle and prove:

$$\boxed{\exists j : 1 \leq j \leq k : |U_t \cap S_{i_j}| \geq \frac{|U_t|}{k}}$$

By contradiction, if $\boxed{\forall j : |U_t \cap S_j| < \frac{|U_t|}{k}}$, the maximum number of elements of U_t that could be covered by $S_{i_1}, S_{i_2}, \dots, S_{i_k}$ would be:

$$\boxed{\leq \sum_{j=1}^{k} |U_t \cap S_{i_j}| < k \frac{|U_t|}{k} = |U_t|}$$

contradicting the hypothesis that $S_{i_1}, S_{i_2}, \dots, S_{i_k}$ cover U_t!

Due to the greedy choice A_S_C selects $\boxed{S_t = \arg\max\{|T \cap U_t| : T \in \mathcal{S}\}}$.

Thus $\boxed{|U_t \cap S_t| \geq |U_t \cap S_{i_j}| \geq |U_t|/k}$

Morale: Thanks to the greedy choice I always cover a fraction $\geq \frac{1}{k} = \frac{1}{|\mathcal{C}^*|}$ of the yet uncovered elements!

This gives the important relation between the returned solution and the optimal solution.

We can write the following recurrence:

$$\boxed{\begin{cases} |U_1| = n \\ |U_{t+1}| \leq |U_t| - |U_t|/k = |U_t|\left(1-\frac{1}{k}\right) \end{cases}}$$

By unfolding:

$$|U_{t+1}| \leq |U_t|\left(1-\frac{1}{k}\right) \leq |U_{t-1}|\left(1-\frac{1}{k}\right)^2$$
$$\vdots$$
$$\leq |U_{t-i}|\left(1-\frac{1}{k}\right)^{i+1}$$
$$\vdots$$
$$\underset{\boxed{i=t-1}}{\leq} |U_1|\left(1-\frac{1}{k}\right)^t = n\left(1-\frac{1}{k}\right)^t$$

uncovered elements after iteration t

Thus there cannot be more than

$$\boxed{n\left(1-\frac{1}{k}\right)^{t} < n\cdot e^{-t/k}} \quad \boxed{\left(1-\frac{1}{n}\right)^{n} < e^{-1}}$$

uncovered elements after iteration t.

Set $\boxed{\bar{t} = k\lceil \ln n \rceil}$. We have that

$$\boxed{|U_{\bar{t}+1}| < n\cdot e^{-\bar{t}/k} = n\cdot e^{-k\lceil \ln n\rceil/k} = n\cdot e^{-\lceil \ln n \rceil} \leq n\cdot \frac{1}{n} = 1}$$

Thus: $\boxed{U_{\bar{t}+1} = \emptyset}$. Therefore

$$\boxed{|\mathcal{C}| \leq \bar{t} = k\lceil \ln n\rceil = |\mathcal{C}^*|\lceil \ln n \rceil}$$

Finally:

$$\boxed{\rho(|X|) = \rho(n) \leq \frac{|\mathcal{C}|}{|\mathcal{C}^*|} \leq \frac{|\mathcal{C}^*|\lceil \ln n\rceil}{|\mathcal{C}^*|} = \lceil \ln n \rceil}$$

The quality of the approximation provided by A_S_C$(X,\mathcal{F})$ decreases as $n=|X|$ increases!