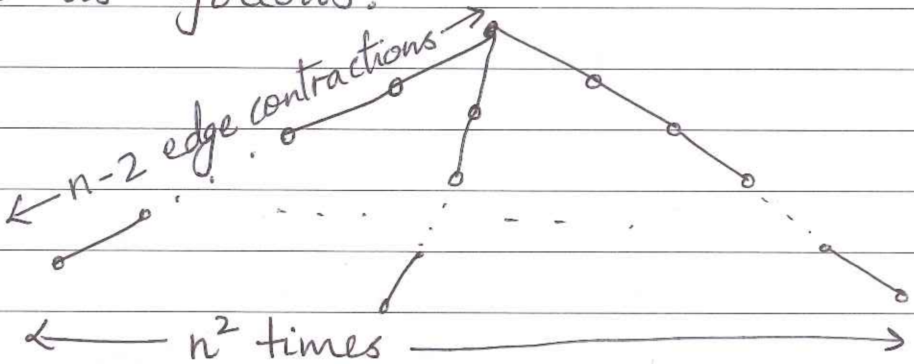
Lecture 5

Recall the randomized global min-cut algorithm.
Its running time is $O(n^4)$ and this algorithm can be visualized as follows:

Here each path denotes the basic algorithm which performed $n-2$ edge contractions.



The success probability of the basic algorithm is quite high in the first few iterations. More precisely,

$$\Pr[\text{no edge of } C \text{ gets contracted in the first iteration}] \geq \frac{n-2}{n}.$$

In the second iteration, we know this probability is $\geq \frac{n-3}{n-1}$ (which is less than $\frac{n-2}{n}$) and it keeps steadily decreasing in every iteration and finally, in the last iteration, it is $\geq \frac{1}{3}$.

Suppose we run the basic algorithm for $n-t$ iterations (for some t). That is, we stop the for loop when the number of remaining vertices is t. What is the probability that no edge of C got contracted in these $n-t$ iterations?

$$\Pr\left[\begin{array}{c}C \text{ survives these} \\ n-t \text{ edge contractions}\end{array}\right] \geq \left(\frac{n-2}{n}\right)\cdot\left(\frac{n-3}{n-1}\right)\cdots\left(1-\frac{2}{n-(n-t)+1}\right)$$

$$= \left(\frac{n-2}{n}\right)\cdot\left(\frac{n-3}{n-1}\right)\cdots\left(\frac{t}{t+2}\right)\cdot\left(\frac{t-1}{t+1}\right)$$

$$= \frac{t\cdot(t-1)}{n(n-1)} \simeq t^2/n^2$$

$\boxed{\simeq \text{ denotes approximately equal to}}$

In our algorithm $t = 2$. Hence our success probability $\geq \frac{2}{n(n-1)}$ — we are referring to the basic algo. here ~~to the basic~~

Suppose we did not run the basic algo. for $n-2$ iterations but only for $n-t$ iterations, for some larger t.

- Then we are left with a graph G_{n-t} on t vertices (these are "super vertices") and with probability $\geq t^2/n^2$, our min-cut C is preserved in G_{n-t}.

How do we find a min-cut in G_{n-t}? We will use the algorithm from last lecture that finds a min-cut with probability $\geq 3/4$ and runs in $O(t^4)$ time in G_{n-t} since this graph has t vertices.

What should t be so that our running time is minimized?

- running the basic algorithm for $n-t$ iterations takes $n(n-t)$ time.

So the running time now is $n(n-t) + t^4$. Let us try to balance n^2 and t^4, i.e., set $n^2 = t^4$. So $t = \sqrt{n}$.

So we have a revised basic algorithm whose running time is $O(n^2)$ and whose success prob. $\geq \underbrace{\frac{t^2}{n^2}}_{\text{Prob. that } C \text{ is preserved in } G_{n-t}} \cdot \underbrace{\frac{3}{4}}_{\text{Prob. that the } O(t^4) \text{ algo. finds a min-cut in } G_{n-t}} = \frac{3}{4n}$ ← Note that we have improved this from $2/n^2$

In order to make the success probability $\geq 3/4$, let us repeat the revised basic algorithm $\left\lceil \frac{8n}{3} \right\rceil$ times and take the least-sized candidate min-cut.

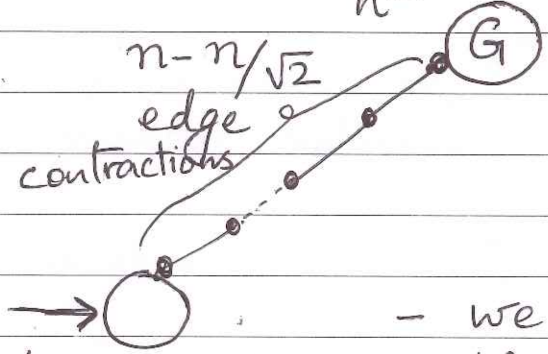
So the total running time is $O(n^3)$ and please check that the error probability $\leq 1/4$.

Question: Can we further improve the running time by using recursion?

How small can we make t so that we can say C survives in the resulting graph (after $n-t$ edge contractions) with probability $\geq \frac{1}{2}$?

That is, we want $\frac{t^2}{n^2} \geq \frac{1}{2}$.

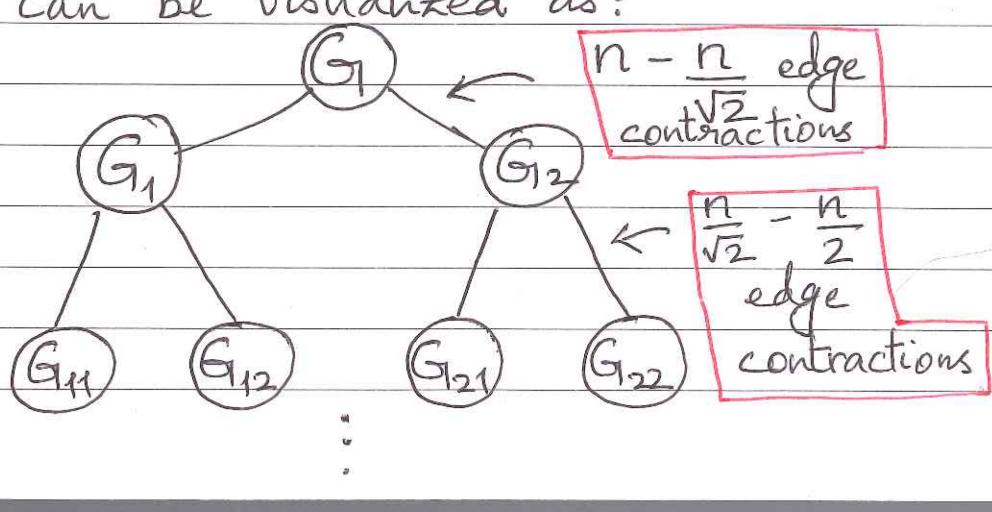
So $t \geq \frac{n}{\sqrt{2}}$



C survives in this graph with probability $\frac{1}{2}$

Idea: At this point repeat the whole process again independently.
– we expect one of these two attempts will retain C. Which one?

So the overall algorithm can be visualized as:



* recursively find min-cut in each of these 2 graphs and find out.

We are now ready to write down the following recursive algorithm.

Date

Randomized - mincut (G, n)

1. Set $t = \left\lceil \frac{n}{\sqrt{2}} + 1 \right\rceil$.

2. Perform n-t random edge contractions on G to get a graph G_1 on t vertices.

3. Call Randomized-mincut (G_1, t). Let C_1 be the candidate min-cut returned by this algorithm.

4. Perform independently another round of n-t random edge contractions on G to get a graph G_2 on t vertices.

5. Call Randomized-mincut (G_2, t). Let C_2 be the candidate min-cut returned by this algorithm.

6. Return the smaller of C_1, C_2.

Note that $\lceil 6/\sqrt{2} + 1 \rceil = 6$. So when $n = 6$, $t = n$.

So we need to have "Step 0" where we say:
- if $n \leq 6$ then compute min-cut in G by brute force.

The recurrence for the running time of the above algorithm is:

$$T(n) = 2 \cdot T\left(\frac{n}{\sqrt{2}}\right) + cn^2 \text{ if } n > 6$$
$$= O(1) \text{ if } n \leq 6.$$

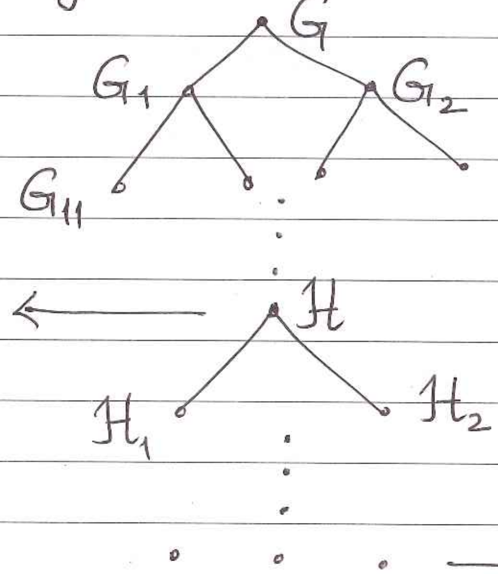
Exercise: Show that $T(n) = O(n^2 \log n)$.

This algorithm is called Karger-Stein algorithm.

Its running time is $O(n^2 \log n)$ which is much better than the previous two algorithms. But what is its success probability?

- if its success probability is low then we have to repeat the algorithm several times and take the best answer so as to amplify its success probability to $\geq 3/4$.

Karger-Stein algorithm can be visualized as follows:



Here each thick point represents a graph where this algorithm is recursively called.

level h

level h-1

Let H be a graph where we find min-cut. That is, our algorithm calls Randomized-mincut(H, r) where r = number of vertices in H.

level 0

here number of vertices ≤ 6. So we will find min-cut by brute force

Note that level of a graph is its height in the binary tree drawn above. What is the height of G?

Recall our favourite min-cut C in G. Suppose no edge of C got contracted in the edge contractions $G \to \cdots \to H$. That is, let us assume that C is preserved in H. Then what is the probability that Randomized-mincut(H, r) finds a min-cut in H?

Let p_h denote this probability, i.e., the probability that Randomized-mincut(H, r) returns a min-cut in H (under the assumption that C is preserved in H).

We know what Randomized-mincut (H, r) does.

$r - r/\sqrt{2}$ edge contractions $\leftarrow$ H $\rightarrow$ $r - r/\sqrt{2}$ edge contractions.

H_1 $\qquad$ H_2

Define the following two events $\mathcal{E}_1$ and $\mathcal{E}_2$:

- let $\mathcal{E}_1$ = the event that min-cut C is preserved in H_1 and the recursive call Randomized-mincut(H_1, r_1) returns a min-cut in H_1

- let $\mathcal{E}_2$ = the event that min-cut C is preserved in H_2 and Randomized-mincut(H_2, r_2) returns a min-cut in H_2.

Observe that $p_h \geq \Pr(\mathcal{E}_1 \cup \mathcal{E}_2)$.

$$\Pr(\mathcal{E}_1 \cup \mathcal{E}_2) = \Pr(\mathcal{E}_1) + \Pr(\mathcal{E}_2) - \Pr(\mathcal{E}_1 \cap \mathcal{E}_2)$$

$$= \Pr(\mathcal{E}_1) + \Pr(\mathcal{E}_2) - \Pr(\mathcal{E}_1) \cdot \Pr(\mathcal{E}_2).$$

$$\Pr(\mathcal{E}_1) = \underbrace{\Pr(C \text{ is preserved in } H_1)}_{\text{this is } \geq 1/2} \cdot \underbrace{\Pr\left(\text{Randomized-mincut}(H_1, r_1) \text{ returns a min-cut} \mid C \text{ is preserved in } H_1\right)}_{\text{this is } p_{h-1}}$$

Hence $p_h \geq \frac{1}{2} \cdot p_{h-1} + \frac{1}{2} \cdot p_{h-1} - \frac{1}{4} p_{h-1}^2$

$$= p_{h-1} - \frac{p_{h-1}^2}{4}; \quad \text{Also } p_0 = 1$$

$\boxed{\text{We will solve this in next lecture.}}$