Eksamensdisposition - Kapitel 1

Søren Mulvad, rbn601

17. juni 2019

- **QuickSort**
  - Analyse af forventet køretid
- **Min-Cut**
  - Håndkørsel
  - Analyse af sandsynligheden for korrekt svar

Eksamensdisposition - Kapitel 1

Randomized QuickSort (RQS)

Vi starter med at definere "rank" i af et element, som vi lader være dets position $S_{(i)}$ i det endelige sorterede array (1-indeksering), altså:

$S_{(i)}$: Element med rank i (givet 1-indeksering)

F.eks.:

```
input:  [3, 0, 5, 2] (usorteret array)
output: [0, 2, 3, 5] (sorteret array)
```

Så vil eksempelvis $S_{(2)} = 2$ og $S_{(4)} = 5$

For alle $1 \leq i < j \leq n$ har vi, at $X_{ij} = 1$ hvis $S_{(i)}$ og $S_{(j)}$ sammenlignes af RQS, og 0 ellers. Vi kan gøre det med en simpel indikatorvariabel, da to elementer aldrig kan blive sammenlignet mere end én gang.

Da får vi følgende, hvor startgrænsen $j > i$ på det andet sumtegn gælder da sammenligningerne er symmetriske.

$$X = \sum_{i=1}^{n} \sum_{j>i} X_{ij}$$

Mål: Find øvre grænse for $\mathbb{E}[X]$
Nu bruges linearity of expectation:

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} \sum_{j>i} X_{ij}\right] = \sum_{i=1}^{n} \sum_{j>i} \mathbb{E}[X_{ij}] = \sum_{i=1}^{n} \sum_{j>i} \mathbb{P}[X_{ij} = 1] \tag{1}$$

Herefter analyserer vi algoritmen:
Den fungerer på følgende måde, at hvis vi f.eks. vælger $S_{(4)}$ som pivot i et array med seks elementer vil vi have:

$$[S_{(1)}, S_{(2)}, S_{(3)},] \underbrace{S_{(4)}}_{\text{pivot}}, [S_{(5)}, S_{(6)}] \tag{2}$$

Herefter vil $S_{(4)}$ aldrig blive sammenlignet med nogle elementer igen, og algoritmen ville nu kigge på de to delarrays der er lavet hver for sig.
Lad os skrive det op mere generelt, hvor vi har to indekser i, j hvor $1 \leq i < j \leq n$. Vi kan da skrive vores array op således:

$$S_{(1)}, S_{(2)}, S_{(3)}, \cdots, S_{(i)}, \cdots, S_{(j)}, \cdots, S_{(n-1)}, S_{(n)}$$

Nu vælges en pivot $S_{(k)}$. Da har vi tre mulige cases som beskrevet nedenfor for hvor k vælges:

1. $k < i$ eller $k > j$:
   $S_{(i)}$ og $S_{(j)}$ bliver ikke sammenlignet i denne iteration af rekursionstræet

2. $i < k < j$:
   $S_{(i)}$ og $S_{(j)}$ sammenlignes *aldrig*. Det skyldes at de vil blive delt ind i hver deres subarray, se evt. (2).

3. $k = i$ eller $k = j$:
   $S_{(i)}$ og $S_{(j)}$ sammenlignes netop én gang (og herefter aldrig igen).

Vi kan nu i vores analyse ignorere case 1, da $S_{(i)}$ og $S_{(j)}$ ikke sammenlignes her, men vi stadig går et niveau dybere i vores rekursionstræ og der derfor er mulighed for at de sammenlignes senere.

Vi ser generelt for case 2 og 3, at de gælder når $i \leq k \leq j$. Case 3 giver os 2 mulige udfald for at de bliver sammenlignet. Antallet af mulige udfald i alt er længden fra i til j, som må være $j - i + 1$. Altså får vi:

$$\mathbb{P}\left[X_{ij} = 1\right] = \frac{2}{j - i + 1} \tag{3}$$

Vi benytter nu dette til at regne videre på (1):

$$\begin{aligned}
\mathbb{E}\left[X\right] &= \sum_{i=1}^{n} \sum_{j>i} \mathbb{P}\left[[] \, X_{ij} = 1\right] \\
&= \sum_{i=1}^{n} \sum_{j>i} \frac{2}{j - i + 1} \\
&= 2 \sum_{i=1}^{n} \sum_{j>i} \frac{1}{j - i + 1} \\
&\leq 2 \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{1}{k} && (4) \\
&= 2n \sum_{k=1}^{n} \frac{1}{k} \\
&= 2nH_n && (5) \\
&= O(n \lg n)
\end{aligned}$$

Ovenfor gælder uligheden ved (4), da $j - i + 1 \geq 2$ (da j altid minimum er 1 større end i), og j ikke vil gå "længere ud" end til n.
I (5) benytter vi hvad der kaldes det n'te Harmoniske tal, som er defineret som $H_n = \sum_{i=1}^{n}(1/i) = \ln n + \Theta(1)$.

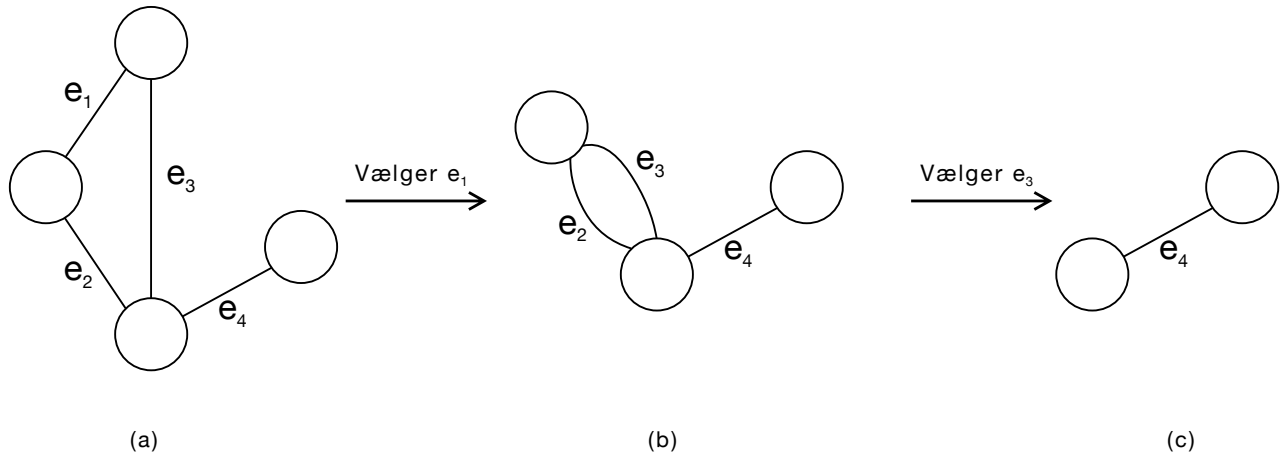
Herved kommer vi altså frem til, at $\mathbb{E}\left[X\right] = O(n \lg n)$. Da $\mathbb{E}\left[X\right]$ er det forventede antal sammenligninger som Randomized QuickSort udfører, må køretiden ligeledes være dette for algoritmen.

Forklaring af algoritmen

Givet en sammenhængende ikke-orienteret graf $G = (V, E)$, da er et min-cut i G en kantmængde E' af minimal størrelse, så $(V, E \backslash E')$ er ikke-sammenhængende.

Algoritmen fungerer således:

- Vælg kant e uniformt i E
- Contract e og fjern self-loops
- Gentag indtil antal knuder $|V'| = 2$
- Returner kantmængde mellem de to knuder



Figur 1: Eksempel på Min-Cut

På Figur 1 fra (a) til (b) vælger vi e_1. Herved sker der en contraction. Fra (b) til (c) vælger vi e_3, hvorved der igen sker en contraction. Vi ser, at vi nu har et self-loop, så det fjerner vi. Nu er der kun to knuder tilbage, så algoritmen er færdig og vi returnerer kantmængden $\{e_4\}$.

NB: Bemærk at vi fra (b) til (c) potentielt godt kunne have valgt kant e_4, og herved ville vi ikke have fundet et min-cut.

Sandsynlighed for succes for én iteration

Lad os nu betragte et min-cut C og lade k være antallet af kanter i C, $k = |C|$. Lad os derudover sige, at $n = |V|$. Og lad os indføre begrebet "degree" for hver knude, som beskriver antal kanter der rører knuden.

Betragt iteration i hvor $1 \leq i \leq n-2$ (dette er netop det antal iterationer algoritmen vil gå igennem, da vi stopper når der er to knuder tilbage og der fjernes en knude pr. contraction).

Antag, at ingen af de kanter der løbende blev contracted i G i iteration $1 \,..\, i-1$ indgår i vores optimale løsning C, således vi ikke potentielt får en fejlagtig løsning.

Da der er n knuder til at starte med og der fjernes én pr. contraction, så må der efter $i-1$ iterationer være $n-i+1$ knuder i G.

Da $k = |C|$ for et min-cut C, så må minimum cut-størrelsen i G være $\geq k$. Derfor må der altså også som minimum være k kanter der rører hver knude, så $G_{\text{min\_degree}} \geq k$.

Da vi havde, at der var $n-i+1$ knuder i G, og vi lige har vist at de alle minimum har degree k, så kan vi få et lower bound for antal kanter $|E|$ i G:

$$|E| \geq \frac{(n-i+1)k}{2}$$

Her dividerer vi med 2, da vi tager højde for at enhver kant e vil røre to knuder.

Således kan vi bestemme sandsynligheden for at en af kanterne i C contractes i iteration i til:

$$\mathbb{P}\left[\text{En kant i } C \text{ contractes i iteration } i\right] = \frac{k}{|E|} \leq \frac{k}{\frac{1}{2}(n-i+1)k} = \frac{2}{n-i+1} \tag{6}$$

Dette gælder, da k er antallet af kanter i C og vi ønsker at finde sandsynligheden for at vi tilfældigt vælger en af disse kanter ud af alle kanter $|E|$.

Lad os nu definere eventet $\mathcal{E}_i$, som svarer til "det omvendte" af eventet beskrevet i (6)
$\mathcal{E}_i$: Ingen kant indeholdt i C vælges i iteration i.

Sandsynlighed for $\mathcal{E}_i$ givet succes i alle foregående iterationer kan herved beskrives som:

$$\mathbb{P}\left[\mathcal{E}_i \middle| \bigcap_{j=1}^{i-1} \mathcal{E}_j\right] \geq 1 - \frac{2}{n-i+1} = \frac{n-i+1-2}{n-i+1} = \frac{n-i-1}{n-i+1} \tag{7}$$

Sandsynlighed for succes for én hel kørsel

Da sandsynligheden for at algoritmen terminerer med korrekt resultat svarer til sandsynligheden for at ingen kanter i C vælges i *nogle som helst* af iterationerne (hvor vi tager højde for at der muligvis kan findes flere gyldige min-cut), kan vi benytte (7) til at opskrive følgende:

$$\begin{aligned}
\mathbb{P}\left[\text{Succes}\right] &\geq \mathbb{P}\left[\bigcup_{i=1}^{n-2} \mathcal{E}_i\right] && (8)\\
&= \mathbb{P}\left[\mathcal{E}_1\right] \cdot \mathbb{P}\left[\mathcal{E}_2|\mathcal{E}_1\right] \cdot \mathbb{P}\left[\mathcal{E}_3|\mathcal{E}_1 \cap \mathcal{E}_2\right] \cdots \mathbb{P}\left[\mathcal{E}_{n-2} \middle| \bigcap_{i=1}^{n-3} \mathcal{E}_i\right] \\
&= \prod_{i=1}^{n-2} \mathbb{P}\left[\mathcal{E}_i \middle| \bigcap_{j=1}^{i-1} \mathcal{E}_j\right] \\
&\geq \prod_{i=1}^{n-2} \left(\frac{n-i-1}{n-i+1}\right) && (9)\\
&= \frac{\cancel{(n-2)\cdots 4 \cdot 3} \cdot 2 \cdot 1}{n(n-1)\cancel{(n-2)\cdots 4 \cdot 3}} \\
&= \frac{2}{n(n-1)} \\
&\geq \frac{2}{n^2}
\end{aligned}$$

I (8) gælder ulighedstegnet, da der muligvis findes flere gyldige min-cuts.
I (9) indsætter vi vores resultat fra (7).

Sandsynlighed for succes givet flere kørsler af algoritmen

Vi har netop set for den randomiserede min-cut algoritme, at for én kørsel på et input med $n = |V|$, antal knuder, har vi:

$$\mathbb{P}\left[\text{Succes}\right] \geq \frac{2}{n^2} \quad \Longleftrightarrow \quad \mathbb{P}\left[\text{Fejl}\right] \leq 1 - \frac{2}{n^2}$$

Derudover ved vi, at følgende ulighed gælder for alle $x \in \mathbb{R}$:

$$1 + x \leq e^x \tag{10}$$

Lad os nu forestille os, at vi kører Min-Cut k gange og returnerer det mindste cut fundet. Denne fremgangsmåde kan kun fejle hvis ingen af kørslerne finder et min-cut. Lad os for at gøre vores analyse nemmere sige vi udfører $k = n^2/2$ kørsler.

$$\begin{aligned}
\mathbb{P}\left[\text{Fejl med } k \text{ kørsler}\right] &\leq \left(1 - \frac{2}{n^2}\right)^k && (11)\\
&\leq \left(e^{-\frac{2}{n^2}}\right)^k && (12)\\
&= e^{-\frac{2k}{n^2}} && (13)\\
&= e^{-1} && (14)\\
&\Updownarrow \\
\mathbb{P}\left[\text{Succes med } k \text{ kørsler}\right] &\geq 1 - e^{-1}
\end{aligned}$$

I (11) ganger vi blot sandsynlighederne for at få fejl k gange sammen.
I (12) benytter vi reglen i (10), hvor $x = -2/n^2$.
I (13) bruger vi potensregler til at sætte k ind i samme potens.
I (14) benytter vi, at vi sagde vi udførte $k = n^2/2$ kørsler.