Algoritmer og Datastrukturer 2 (Sommeren 2007)

Opgave 1

Spørgsmål a:

Antal knuder: $n = k \cdot \ell$.

Antal kanter: $m = k \cdot (\ell - 1) + \ell \cdot (k - 1) = 2\ell k - \ell - k$.

Dijkstra's algoritme: $O(k\ell \log(k\ell))$.

Spørgsmål b:

Korteste vej fra P til E:

$$P \rightarrow Q \rightarrow R \rightarrow \hat{S} \rightarrow \hat{N} \rightarrow O \rightarrow J \rightarrow E$$

Vægt: $2 + 4 + 3 + 2 + 2 + 3 + 1 = 17$

Spørgsmål c:

Korteste vej uden to sving:

$$P \rightarrow Q \rightarrow R \rightarrow \hat{S} \rightarrow N \rightarrow I \rightarrow \hat{D} \rightarrow C \rightarrow \hat{B} \rightarrow G \rightarrow \hat{L} \rightarrow M \rightarrow N \rightarrow \hat{O} \rightarrow J \rightarrow E$$

Vægt: $2 + 4 + 3 + 2 + 2 + 4 + 1 + 3 + 2 + 1 + 3 + 4 + 2 + 3 + 1 = 37$

Spørgsmål d:

En løsning er at repræsentere hver knude som 12 knuder, der repræsenterer de sidste to skridt taget man er kommet fra til en givet knude. Knuden H bliver til følgende 12 knuder, hvor pilene indikerer de sidste to skridt taget.

$H_1 :\rightarrow\rightarrow$	$H_2 :\rightarrow\uparrow$	$H_3 :\rightarrow\downarrow$
$H_4 :\leftarrow\leftarrow$	$H_5 :\leftarrow\uparrow$	$H_6 :\leftarrow\downarrow$
$H_7 :\uparrow\uparrow$	$H_8 :\uparrow\rightarrow$	$H_9 :\uparrow\leftarrow$
$H_{10} :\downarrow\downarrow$	$H_{11} :\downarrow\rightarrow$	$H_{12} :\downarrow\leftarrow$

Så knuden H_1 repræsenterer at man er gået to gange til højre, og derfor nu kan gå hvorhen man vil. Lad O være knuden over H, U knuden under H, L knuden til venstre for H, og R knuden til højre for H. Fra H_1 kan man gå til O_2 man kan gå til R_1 og til U_3. Fra knuden H_2 kan man gå til O_7. Fra knude H_3 kan man gå til U_{10}. Og så fremdeles. Hvis den nederste venstre knude betegnes s så findes en kortest sti ved at køre Dijkstra's algoritme med s_1 og s_7 som start knuder. Udførselstid for Dijkstra på denne graf bliver $O(\ell k \log(\ell k))$ da hver knude skaleres til $O(1)$ knuder.

Opgave 2

Spørgsmål a:

$$V_1 = \{A, E, J, K\} \quad V_2 = \{B, C, D, F, G, H, I\}$$

Spørgsmål b:

Vælg en tilfældig knude s. Lav DFS gennemlæøb af grafen startende i s. Lad V_1 være knuder med en lige dybde i DFS træet og V_2 knuder med en ulige dybde i DFS træet. Grafen er todelt hvis og kun hvis der ikke findes en kant der forbinder to knuder med lige dybde i DFS træet eller to knuder med ulige dybde. Løses med DFS i tid $O(m)$.

Opgave 3

Spørgsmål a:

Minimum udspændende træ:

$$(B, C), (A, C), (F, H), (D, F), (B, D), (G, H), (A, I), (J, K), (E, G), (E, J)$$

Vægt: 56

Spørgsmål b:

Givet $G = (V, E)$ og $MST(G)$. Hvis $e \notin MST(G)$ gøres intet. Ellers fjern e fra $MST(G)$. Lav DFS fra en knude u i $MST(G)$ og marker alle knuder der findes med V_1. Lad V_2 være resten. V_1 og V_2 definerer et snit(opdeling). Skan E og find kant e' mellem V_1 og V_2 med mindst vægt. Indsæt e' i $MST(G)$. Udførselstid $O(m)$.

Opgave 4

Spørgsmål a:

Tabel:

$L(i,j)$	0	1	2	3
0	0	1	2	3
1	1	2	3	3
2	2	2	3	4
3	3	3	4	4
4	4	4	4	5

Spørgsmål b:

Følgende pseudkode skulle du:

```
Maketable(S,T)
  for i=0 to m
     for j=0 to n
        if (i = 0) L[i,j]  := j
        else if (j = 0) L[i,j] := i
        else if (x_i != y_j) L[i,j] := 1 + min{L[i-1,j],L[i,j-1]}
        else L[i,j] := 1 + L[i-1,j-1]
  return L[m,n]
```

Udførselstid for Maketable: $O(n \cdot m)$

Spørgsmål c:

Følgende metode tager tabellen L lavet i opgave b: og tallene n, m.

```
Print(L,i,j)
  if (i = 0 and j = 0) return
  if (i = 0) write(y_1,...y_j) return
  if (j = 0) write(x_1,...x_i) return
  if (x _i != y_j)
     if (1 + L[i-1,j] = L[i,j])
        Print(L,i-1,j)
        write(x_i)
     else
        Print(L,i,j-1)
        write(y_j)
  else
     Print(L,i-1,j-1)
     write(x_i)
```

Udførselstid for Print(L,m,n): $O(n + m)$

Samlet udførselstid: $O(n \cdot m)$

Opgave 5

Spørgsmål a:

Suffixerne i sorteret orden:

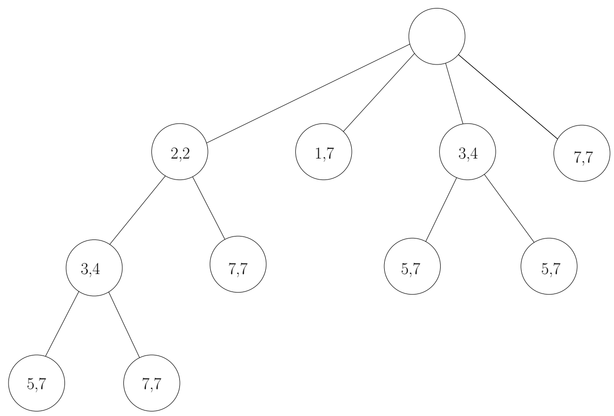
Streng	Start Index
ANANAS	2
ANAS	4
AS	6
BANANAS	1
NANAS	3
NAS	5
S	7

Det giver suffix array:

2	4	6	1	3	5	7

Spørgsmål b:

Suffix-træ:



Spørgsmål c:

To løsninger:

1) Den letteste. Iterer over alle delstrenge af S af længde k. Der er $n-k+1$ af dem. For hver kør Knuth-Morris-Pratt algorithmen og tæl alle forekomster. Returner den der forekommer flest gange. Udførselstid: $O(n^2)$.

2) Den hurtigste. Lav suffixtræ for S i $O(n)$ tid. For alle knuder i træet hvor stien fra roden til og med knuden repræsenterer en streng af længde k eller mere. Tæl antal blade i undertræet. Returner for den knude, der har flest blade under sig, de k første tegn langs stien fra roden til knuden. Alt dette kan klares i et konstant antal lineære gennemløb af træet. Udførselstid $O(n)$.