

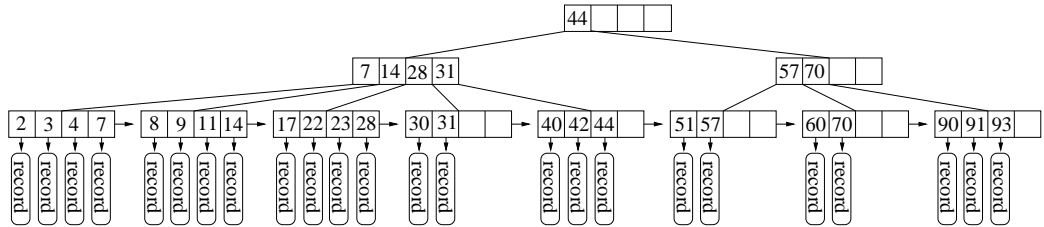
Examen Algoritmen en Datastructuren III

Naam :

Stellingen uit de les mogen natuurlijk altijd gebruikt worden zonder dat het bewijs opnieuw gegeven moet worden (behalve in gevallen waar het expliciet anders staat)!

1. B+-bomen (3 pt)

- Voeg de sleutel 24 toe aan de volgende B+-boom met grootte 4. Toon voldoende tussenstappen en details om te kunnen zien wat er gebeurt.



- Sommige sleutels in een B^+ -boom zitten in bladeren **en** in interne toppen. Toon aan dat als een blad gesplitst wordt en een kopie van een sleutel in de ouder geplaatst wordt die niet al in de ouder aanwezig is. Gebruik **expliciet** de definitie van een B^+ -boom.

- Ontwikkel een formule voor de **minimale** diepte van een B+-tree met grootte n en s sleutels in de bladeren.

2. Hashing (3 pt)

- Voeg sleutels met de volgende hash-waarden (in binaire vorm) in de gegeven volgorde toe aan een extensible hashing tabel waarbij je in het begin maar 1 bit van de hash-waarde gebruikt:

001010, 100100, 000111, 001000, 101111, 111111, 110011

Er mogen twee sleutels in één emmer zitten. Zoals in de les worden er ook hier de hash-waarden in de tabel geschreven in plaats van de sleutels (die toch al niet gegeven zijn). Toon voldoende tussenstappen en details om te kunnen zien wat er gebeurt.

3. Compressie (3 pt)

Stel dat A_C een compressiealgoritme is dat afhankelijk van hoe vaak elk letterteken in een tekst voorkomt voor elk letterteken in het alfabet van de tekst een vaste code gebruikt en de tekst dan codeert door de codes van de lettertekens in dezelfde volgorde op te slaan als de lettertekens in de originele tekst – b.v. (statisch) Huffman coderen.

- Stel dat A_C een compressiealgoritme is dat afhankelijk van hoe vaak elk letterteken in een tekst voorkomt voor elk letterteken in het alfabet van de tekst een vaste code gebruikt en de tekst dan codeert door de codes van de lettertekens in dezelfde volgorde op te slaan als de lettertekens in de originele tekst – b.v. (statisch) Huffman coderen.

Dit examen is in latex geschreven – de source is dus een tekstbestand. Stel dat je het tekstbestand één keer met A_C codeert en één keer eerst de Burrows-Wheeler getransformeerde berekent en op die dan A_C toepast. Wat kan je over de verhouding tussen de lengten van deze twee codes zeggen? Geef uitleg.

- Geef uitleg waarom de Burrows-Wheeler getransformeerde van een tekst (normaal) beter gecomprimeerd kan worden dan de tekst zelf.

- Bereken de Burrows-Wheeler getransformeerde van `gissen_is_mis` en pas op het resultaat de *move to front* codering toe waarbij in het begin alle lettertekens in lexicografische volgorde staan. Als je het eerste deel niet kan oplossen pas de *move to front* codering op de ongewijzigde tekst toe.

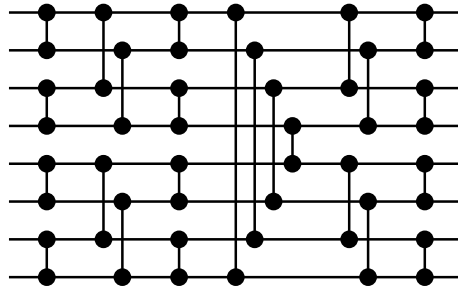
- Codeer de tekst

`"gissen_is_mis"_is_mis`

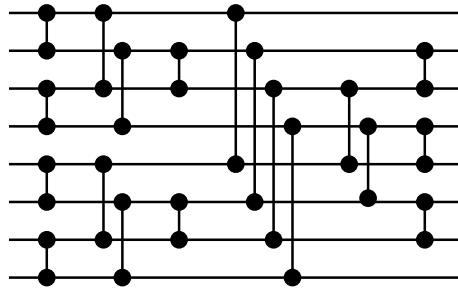
met LZ77 waarbij je een sliding window van grootte 8 gebruikt. Toon voldoende tussenstappen en details om te kunnen zien wat er gebeurt.

4. Sorteernetwerken (3 pt.)

- Sorteert het onderstaande netwerk correct? Zo nee, geef een tegenvoorbeeld. Zo ja, bewijs.



- Gebruik het nul-een-principe voor sorteernetwerken om te bewijzen dat het onderstaande netwerk correct sorteert.



5. Netwerktopologieën (2 pt.)

- Beschouw een ring van p processoren. Veronderstel dat elke processor hoogstens k boodschappen verstuurt en dat elke processor van hoogstens k boodschappen de bestemming is. Bespreek hoe dit communicatieprobleem kan worden opgelost in tijd $\leq \frac{(k+1)p}{2}$.

- Een **cube-connected-cycles**-graaf is een ongerichte kubische graaf, die gevormd wordt door elke top van een hyperkubus te vervangen door een cykel. Deze graaf werd ontworpen als een alternatief met kleine graad voor de hyperkubus.

Formeel wordt de cube-connected-cycles-graaf van orde d (genoteerd als CCC_d) als volgt gedefinieerd. De toppenverzameling heeft $d2^d$ toppen, gelabeld door getalparen (x, y) met $0 \leq x < 2^d$ en $0 \leq y < d$. Elke top (x, y) heeft drie burens, nl. $(x, (y + 1) \bmod d)$, $(x, (y - 1) \bmod d)$ en $(x \text{ XOR } 2^y, y)$, waarbij XOR staat voor de bitsgewijze exclusieve-or op binaire getallen.

Bepaal de diameter van CCC_3 .

6. Exact string matching (2 pt.)

- Geef een voorbeeld van tekst (van lengte n) en patroon (van lengte m) waarvoor het algoritme van Knuth-Morris-Pratt minstens $\Omega(n + m)$ vergelijkingen doet.

- Geef een voorbeeld van tekst (van lengte n) en patroon (van lengte m) waarvoor het algoritme van Boyer-Moore-Horspool minstens $\Omega(nm)$ vergelijkingen doet.

NOG NIET OMDRAAIEN !