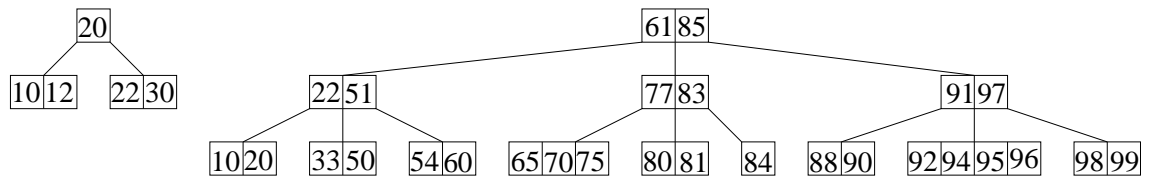


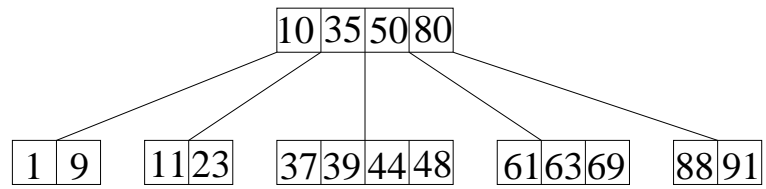
2. (2 pt)

- Wat is het voordeel van een B-boom in vergelijking met een AVL-boom? Wanneer gebruik je B-bomen?

- Schrijf voor elke van de volgende bomen of hij aan de definitie van een B-boom met grootte 4 voldoet. In het geval van niet schrijf waarom niet.



- Voeg sleutel 40 toe aan de volgende B-boom met grootte 4. Toon voldoende tussenschappen om te zien wat er gebeurt.



3. (3 pt)

Jullie kennen het inpakprobleem al uit AD2: Gegeven een reeks van n gewichten g_1, \dots, g_n met $0 < g_i \leq 1$ voor $1 \leq i \leq n$. Het doel is de gewichten zo op vrachtwagens te plaatsen dat de som van de gewichten op elke vrachtwagen ten hoogste 1 is en er zo weinig als mogelijk vrachtwagens gebruikt worden.

Ook dit probleem is NP-compleet dus lijkt het een goed idee voor grote waarden van n een metaheuristiek toe te passen.

- Geef een guided local search algoritme voor dit probleem. Geef alle definities en parameters die nodig zijn om het algoritme volledig te beschrijven maar het is niet nodig dat ze ook zo gekozen zijn dat het algoritme bijzonder goed werkt. Het algoritme moet alleen maar *in principe* werken en het optimum kunnen vinden (ook al zou dat veel te lang duren en te veel pogingen vragen).

De bedoeling is gewoon om aan te tonen dat het principe verstaan is.

- Geef een voorbeeldprobleem en toon hoe jouw algoritme werkt. Het is voldoende als getoond wordt hoe de belangrijke delen van jouw algoritme werken. Het is niet noodzakelijk door te gaan tot het globale optimum is gevonden of het algoritme op de computer zou stoppen!

4. (2 pt)

Stel dat je niet één groot bestand moet sorteren, maar n al bestaande gesorteerde bestanden b_1, \dots, b_n tot één groot gesorteerd bestand moet samenvoegen. Voor $1 \leq i \leq n$ bevat bestand b_i l_i sleutels en je kan de bestanden niet tegelijk mergen maar je kan altijd alleen maar 2 bestanden mergen tot een nieuw bestand. Als de oude bestanden l_i en l_j sleutels bevatten, bevat het nieuwe bestand dus $l_i + l_j$ sleutels en de kost van deze mergebewerking is $l_i + l_j$.

- Geef een voorbeeld dat toont dat de kost om alle bestanden te mergen afhankelijk is van de volgorde waarop je de bestanden mergt.
- Beschrijf een efficiënt algoritme om de optimale (goedkoopste) volgorde te bepalen.
- Wat is de complexiteit van dit algoritme?
- Toon aan dat jouw algoritme inderdaad de optimale volgorde vindt.

TIP: Stellingen en algoritmen uit de les mogen natuurlijk gebruikt worden en het bewijs van de stellingen moet niet herhaald worden !

5. (2 pt)

Gegeven is een array (a_1, \dots, a_n) van elementen, waarbij bepaalde elementen **gemarkeerd** zijn. Je mag veronderstellen dat de markeringen gegeven worden in een bijkomende 0-1-array (ℓ_1, \dots, ℓ_n) , waarbij $\ell_i = 1$ aangeeft dat a_i gemarkeerd is, en $\ell_i = 0$ als a_i niet gemarkeerd is.

Gevraagd is de array (a_1, \dots, a_n) zo te herorganiseren dat de gemarkeerde elementen vooraan komen te staan; daarbij moet de onderlinge positie van de gemarkeerde elementen behouden blijven.

Bijvoorbeeld, voor de array $(12, 43, 25, 64, 37)$, met markeringen $(0, 1, 1, 0, 1)$, wordt de aangepaste array $(43, 25, 37, 12, 64)$ bekomen.

Geef een PRAM algoritme dat dit probleem oplost in $\Theta(\log n)$ parallele tijd op $\Theta(n/\log n)$ processoren.

6. (2 pt)

Een 3-dimensionaal rooster heeft processoren $P_{i,j,k}$ die (behalve aan de rand) telkens verbonden zijn met zes burens $P_{i-1,j,k}$, $P_{i+1,j,k}$, $P_{i,j-1,k}$, $P_{i,j+1,k}$, $P_{i,j,k-1}$, $P_{i,j,k+1}$.

- (a) Geef een algoritme om n waarden te sommeren op een 3-dimensionaal rooster met n processoren. Je mag veronderstellen dat $n = \ell^3$, voor zekere ℓ . Bespreek de uitvoeringstijd $T(n)$ en het werk $W(n)$ van het algoritme.
- (b) Gebruik de techniek van accelerated cascading om een optimaal algoritme voor dit probleem te bekomen. Bespreek de uitvoeringstijd $T(n)$ en het werk $W(n)$ van het algoritme.

7. (3 pt)

- (a) Bereken de verschuivingstabel voor het algoritme van Boyer-Moore-Horspool voor het patroon $P = \text{antenna}$. Gebruik het algoritme voor het zoeken naar het patroon P in de tekst $T = \text{anterrantennar}$. Hoeveel vergelijkingen maakt het algoritme?
- (b) Bereken de verschuivingstabel voor het algoritme van Knuth-Morris-Pratt voor het patroon $P = \text{entente}$. Gebruik het algoritme voor het zoeken naar het patroon P in de tekst $T = \text{tenttentetententen}$. Hoeveel vergelijkingen maakt het algoritme?

NOG NIET OMDRAAIEN !