

1. (a) Tiivistetään merkkijono siten, että siitä poistetaan kaikki välilyönnit, ja muita merkkejä siirretään tarvittava määrä kohti merkkijonon alkua. (3 p.)

```

j = -1;
for (i = 0; i < n; i++)
    if (m[i] != ' ')
    {
        j++;
        if (i != j) m[j] = m[i];
    }
n = j + 1;

```

- (b) Kaikki  $n$  merkkiä käydään läpi kerran.  $\implies$  Aikavaativuus  $O(n)$ . (3 p.)

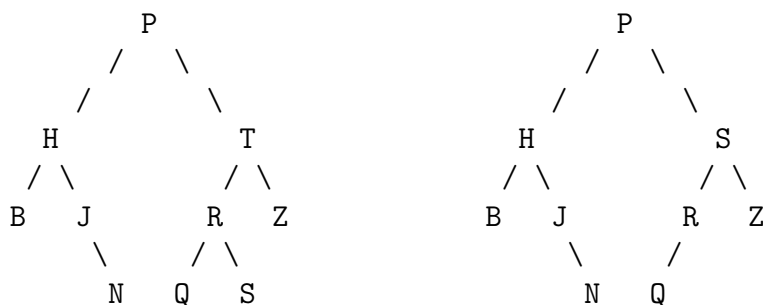
2. (a) Jonon perusoperaatiot: **enqueue** = alkion lisääminen viimeiseksi, **dequeue** = alkion palautus ja poisto alusta. (2 p.)

- (b) Alkiot "kiertävässä" taulukossa  $q[0..QMAX-1]$ , ensimmäisen indeksi  $f$ , viimeistä seuraavan paikan indeksi  $b$ . Operaatio **enqueue**: tarkistus että jono ei ole täysi, sijoitus  $q[b]=alkio$ , muutos  $b=b+1$  tai  $b=0$ . Operaatio **dequeue**: tarkistus että jono ei ole tyhjä, sijoitus  $alkio=q[f]$ , muutos  $f=f+1$  tai  $f=0$ . (2 p.)

- (c) Ensimmäinen ja viimeinen alkio suoraan käytettävissä, operaatioiden suoritus ei riipu alkioden lukumäärästä.  $\implies$  Aikavaativuus  $O(1)$ . (2 p.)

3. (a) Lisätään tyhjään binääripuuhun merkit P, T, H, B, J, R, N, Q, S, Z. (3 p.)

- (b) Poistettavan solmun T paikalle siirretään sen vasemman alipuun rakenteellisessa järjestyksessä viimeinen solmu S. (3 p.)



4. (a) Lomituslajittelu: jaetaan alkiot kahteen osajoukkoon, lajitellaan osajoukot rekursiivisesti, yhdistetään osajoukot lomittamalla. (3 p.)

- (b) Osajoukoissa noin  $n/2$  alkioita, lomitus voidaan tehdä lineaarisesti. Saadaan rekursioyhtälö  $T(n) = 2T(n/2) + cn$ , joka voidaan ratkaista [ratkaisua ei vaadittu].  $\implies$  Aikavaativuus  $O(n \log n)$ . (3 p.)