

Kirjoja, laskinta tms. ei saa olla tentissä. Tehtävällä on yksi tai useampi numero maksimipistemäärän mukaan. Muiden kuin koodaa tai piirrä -tehtävien mallivastaukset ovat melko lyhyet, tyypillisesti 1, ..., 3 riviä pistettä kohti.

1. Anna esimerkki tilanteesta, jossa algoritmeista A ja B kannattaa valita se, joka on Θ -merkinnällä ilmaistuna huonompi.

Syötteet ovat pieniä ja huonompi on pienillä syötteillä tehokkaampi, esimerkiksi INSERTIONSORT vastaan HEAPSORT.

2. Taulukossa $A[0 \dots n-1]$ on kokonaislukuja. Kirjoita pseudokoodi tai ohjelmanpätkä, jonka jälkeen muuttujassa ok on true jos ja vain jos A on keko suurin ensin.

```

1  ok = true;
2  for( i = 1; i < n; ++i ){
3      if( A[ (i-1)/2 ] < A[i] ){ ok = false; break; }
4  }
```

3. Kaupungissa on monta tehdasta ja monta paloasemaa. Halutaan suunnitella reitti kuhunkin tehtaaseen siitä paloasemasta, josta saadaan lyhin reitti. Käytettävissä on Dijkstran reitintensintäalgoritmin toteutus, joka sallii monta maalisolmua mutta vain yhden lähtösolmun. Karttaa voi muuttaa. Miten toteutuksen saa toimimaan monelle paloasemalle?

Lisätään karttaan yksi solmu ja siitä nollan pituiset kaaret kuhunkin paloasemaan.

4. Mitä tarkoittavat binääripuun solmun *korkeus* (*height*) ja *syvyys* (*depth*)?

Korkeus on kaarten määrä mahdollisimman pitkällä polulla solmusta alaspäin. Syvyys on juuresta solmuun vievän polun kaarten määrä.

- 5&6. Tarkastellaan jonkin avaimen esiintymistä useasti tasapainotetussa binäärihakupuussa. Mitkä seuraavista ovat mahdollisia (K, ”kyllä”) ja mitkä mahdottomia (E, ”ei”)? Kukin oikea vastaus $+\frac{1}{2}$ p, väärä $-\frac{1}{2}$ p, tyhjä 0 p. Yhteensä min 0 p. ja max 2 p.

- (a) Se esiintyy juuressa ja vasemmassa alipuussa, mutta ei oikeassa alipuussa.
- (b) Se esiintyy juuressa ja oikeassa alipuussa, mutta ei vasemmassa alipuussa.
- (c) Se esiintyy molemmissa alipuissa mutta ei juuressa.
- (d) Se esiintyy sekä molemmissa alipuissa että juuressa.

(a) K (b) K (c) E (d) K

7. Valitse jokin edellisen tehtävän vastauksistasi (a), (b), (c) tai (d), ja perustele se.

Jos se esiintyy vasemmassa alipuussa niin juuren avain on ainakin yhtäsuuri, ja jos se esiintyy oikeassa alipuussa niin juuren avain on enintään yhtäsuuri, joten juuren avain on täsmälleen yhtäsuuri. Siksi (c) on mahdoton.

- 8&9. Taulukko [1108, 2105, 2026, 4211] järjestetään kantalukujärjestämällä (radix sort). Avaimen osina ovat luvun yksittäiset numerot. Esimerkiksi 2026 sisältää neljä osaa: 2, 0, 2 ja 6. Missä järjestyksessä taulukko on (a) ensimmäisen (b) toisen (c) 3. kierroksen jälkeen?

[4211, 2105, 2026, 1108] [2105, 1108, 4211, 2026] [2026, 2105, 1108, 4211]

Graafista on valittu yksi solmu lähtösolmuksi. Solmun *etäisyys* on mahdollisimman pieni määrä kaaria, joita pitkin siihen pääsee lähtösolmusta. Mitä voidaan päätellä solmun *v* etäisyydestä seuraavissa tapauksissa?

10. Suunnatun graafin solmun *u* etäisyys on *d* ja solmusta *u* on kaari solmuun *v*.

Se on enintään $d + 1$.

11. Suunnatun graafin solmun *u* etäisyys on *d* ja solmusta *v* on kaari solmuun *u*.

Se on vähintään $d - 1$.

12. Suuntaamattoman graafin solmun *u* etäisyys on *d* ja solmujen *u* ja *v* välillä on kaari.

Se on $d - 1$, d tai $d + 1$.

Taulukossa $P[0 \dots n - 1]$ on luvut $0, \dots, n - 1$ jossakin järjestyksessä. Jos aloitetaan mistä tahansa $0 \leq i < n$ ja lasketaan $P[i], P[P[i]], P[P[P[i]]]$ ja niin edelleen, niin lopulta tullaan takaisin lukuun *i*. Nämä luvut yhdessä muodostavat *silmukan*. Esimerkiksi taulukossa $[2, 3, 1, 0]$ on yksi silmukka $0 \rightsquigarrow 2 \rightsquigarrow 1 \rightsquigarrow 3 \rightsquigarrow 0$, ja taulukossa $[0, 1, 2, 3]$ on neljä silmukkaa.

13. Kirjoita taulukko, jossa on neljä alkiota ja kaksi silmukkaa.

$[0, 2, 3, 1]$

14. Perustele, että luku johon tullaan takaisin, on nimenomaan *i*.

Olkoon luvun $0 \leq k < n$ edeltäjä mikä tahansa $0 \leq h < n$, jolle $P[h] = k$. Koska kukin luvusta $0, \dots, n - 1$ on *P*:ssä vain kerran, ei luvulla voi olla kahta eri edeltäjää. Jos ensimmäinen luku *j*, johon tullaan takaisin, olisi muu kuin *i*, niin sillä olisi kaksi eri edeltäjää: se jonka kautta *j* löydettiin ensimmäisen kerran ja se jonka kautta *j* löydettiin jälkimmäisen kerran.

15. Kerro sanallisesti, miten voidaan laskea, kuinka monta silmukkaa on *P*:ssä.

Merkitään kohta 0 löydetyksi. Merkitään myös kohdat $P[0], P[P[0]]$ ja niin edelleen löydetyiksi kunnes ollaan takaisin kohdassa 0. Siinä on yksi silmukka. Jos yhä on merkittävättömiä kohtia, valitaan mikä tahansa niistä ja tehdään sama. Siinä on yksi silmukka lisää. Tätä jatketaan kunnes jokainen kohta on merkitty.

16&17. Kirjoita ohjelmapätkä, joka laskee, kuinka monta silmukkaa on *P*:ssä.

```

1  unsigned silmukoita = 0; bool kohdattu[ n ] = {};
2  for( unsigned i = 0; i < n; ++i ){
3      if( !kohdattu[ i ] ){
4          ++silmukoita; unsigned j = i;
5          while( !kohdattu[ j ] ){ kohdattu[ j ] = true; j = P[j]; }
6      }
7  }
```

Hajautustaulussa on n alkioita ja kantataulukon koko on M . Osoitin eli linkki vie 8 tavua ja hyötykuorma h tavua. Kurssilla kantataulukon alkioit olivat pelkkiä osoittimia, mutta vaihtoehtoisesti niissäkin voi olla myös hyötykuorma. Tyhjän alkion hyötykuorman tilalla on $\sim 0u$.

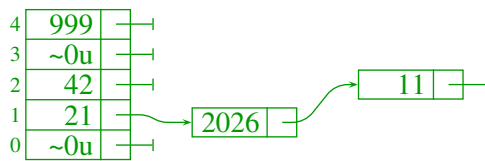
18. Kuinka paljon muistia vie kurssin hajautustaulu ja kuinka paljon vaihtoehtoinen hajautustaulu, jos täsmälleen neljäsosa listoista on tyhjiä?

$$8M + n(h + 8) \text{ ja } (\frac{1}{4}M + n)(h + 8)$$

19. Millä $h:n$, $n:n$ ja $M:n$ arvoyhdistelmillä vaihtoehtoinen hajautustaulu vie vähemmän muistia kuin kurssin hajautustaulu, jos täsmälleen neljäsosa listoista on tyhjiä?

$$(\frac{1}{4}M + n)(h + 8) < 8M + n(h + 8) \Leftrightarrow \frac{1}{4}M(h + 8) < 8M \Leftrightarrow h + 8 < 32 \Leftrightarrow h < 24$$

20. Piirrä vaihtoehtoinen hajautustaulu, kun hajautusfunktiona on $x \bmod 5$ ja alkioit ovat 21, 999, 2026, 42 ja 11.



Seuraavissa tehtävissä tarkastellaan oheista ohjelmanpätkää. Se laskee, kuinka monta muutosta tarvitaan muuttamaan merkkijono A merkkijonoksi B. Esimerkiksi talosta tulee pallo kahdella muutoksella: talo palo pallo. Funktio `std::min(x, y)` palauttaa pienemmän luvusta x ja y . Olkoot A:n pituus n ja B:n pituus m . Tarkoittakoon $X[i \dots j]$ merkkijonon X merkkejä kohdasta i alkaen kohtaan j saakka. Esimerkiksi jos $X = \text{pituus}$, niin $X[1 \dots 3] = \text{itu}$.

```

1  std::vector< unsigned > paras( A.size() + 1 );
2  for( unsigned i = 0; i <= A.size(); ++i ){ paras[i] = i; }
3  for( unsigned j = 1; j <= B.size(); ++j ){
4      unsigned vanha = paras[0]; paras[0] = j;
5      for( unsigned i = 1; i <= A.size(); ++i ){
6          unsigned uusi = paras[i];
7          if( A[i-1] == B[j-1] ){ paras[i] = vanha; vanha = uusi; continue; }
8          paras[i] = std::min( paras[i], vanha );
9          paras[i] = std::min( paras[i], paras[ i-1 ] );
10         ++paras[i]; vanha = uusi;
11     }
12 }
```

21. Ilmaise ajan kulutus hitaimmillaan Θ -, O -, Ω - tms. merkinnällä.

$$\Theta(nm)$$

22. Ilmaise ajan kulutus nopeimmillaan Θ -, O -, Ω - tms. merkinnällä.

$$\Theta(nm)$$

23. Ilmaise lisämuistin kulutus enimmillään Θ -, O -, Ω - tms. merkinnällä. Ilmaise lisämuistin kulutus vähimmillään Θ -, O -, Ω - tms. merkinnällä.

$$\text{Molemmat ovat } \Theta(n).$$

24. Perustele jokin edellä ilmoittamasi ajan kulutus.

Rivien 3, ..., 12 silmukka kierretään m kertaa ja kullakin niistä rivi 5 suoritetaan $n + 1$ kertaa. Niistä tulee $m(n + 1)$, joka on $\Theta(nm)$. Mitään riviä ei suoriteta sen enempää.

25. Perustele jokin edellä ilmoittamasi lisämuistin kulutus.

Rivillä 1 varataan taulukko kooltaan $\Theta(n)$. Muuta paljon lisämuistia kuluttavaa ohjelmanpätöksessä ei ole.

26. Miten huonoimman tapauksen lisämuistin kulutusta voi jonkin verran parantaa, jos A ja B ovat kovin eripituiset?

Jos B on lyhyempi kuin A, vaihdetaan ne aluksi keskenään.

27. Missä vastaus on, kun ohjelmanpätkä on lopettanut?

`paras[A.size()]`

28. Kirjoita oheisen taulukon $x:n$, $y:n$, $z:n$ ja $t:n$ tilalle kuuluvat tiedot.

$x = 3, y = 1, z = 4$ ja $t = 2$

29. Rivin 7 alussa uusi sisältää tiedon, kuinka monta muutosta tarvitaan muuttamaan eräs merkkijono erääksi merkkijonoksi. Mitkä ovat nämä kaksi merkkijonoa?

$A[0 \dots i - 1]$ ja $B[0 \dots j - 2]$

30. Jos `paras[i]` muuttuu rivillä 9, niin minkä muutoksen ohjelmanpätkä valitsi?

$A[i - 1]$ poistetaan

	p	a	l	l	o	
	0	1	2	3	4	5
t	1	1	2	3	4	5
a	2	2	1	2	3	4
l	3	x	2	y	2	3
o	4	z	3	t	2	2

Loppu