Soal Latihan Algoritma Greedy

IF2211 Strategi Algoritma

2023

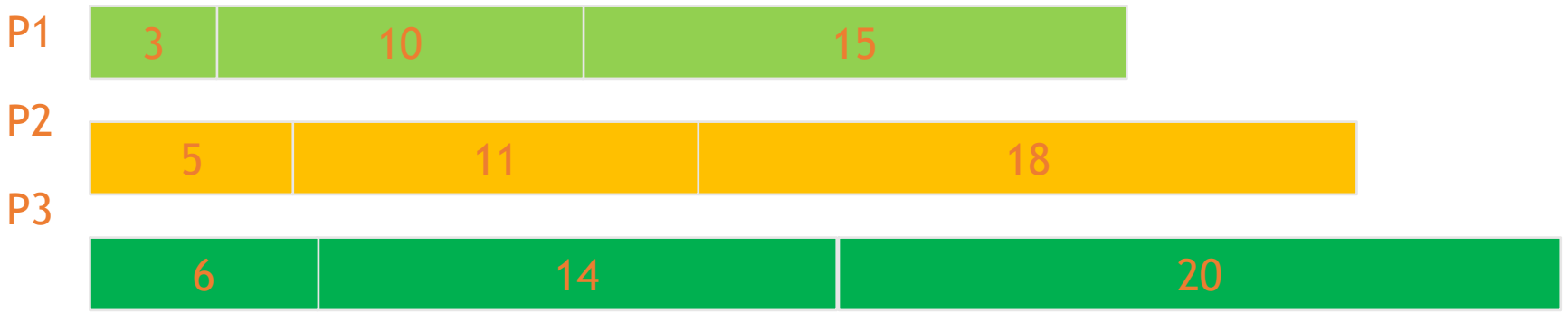
Soal Latihan 1

Misalkan terdapat 9 buah job dengan waktu pengerjaan adalah 3, 5, 6, 10, 11, 14, 15, 18, dan 20 menit. Anda memiliki tiga buah prosesor untuk menjalankan job-job tersebut. Bagaimana cara pengaturan job pada setiap prosesor sehingga waktu penyelesaian semua job adalah minimal? Waktu penyelesaian job dihitung dari prosesor yang terlama menyelesaikan job. Misalkan cara pengaturan job adalah sbb:

Waktu penyelesaian job = 18 + 11 + 6 = 35 menit.

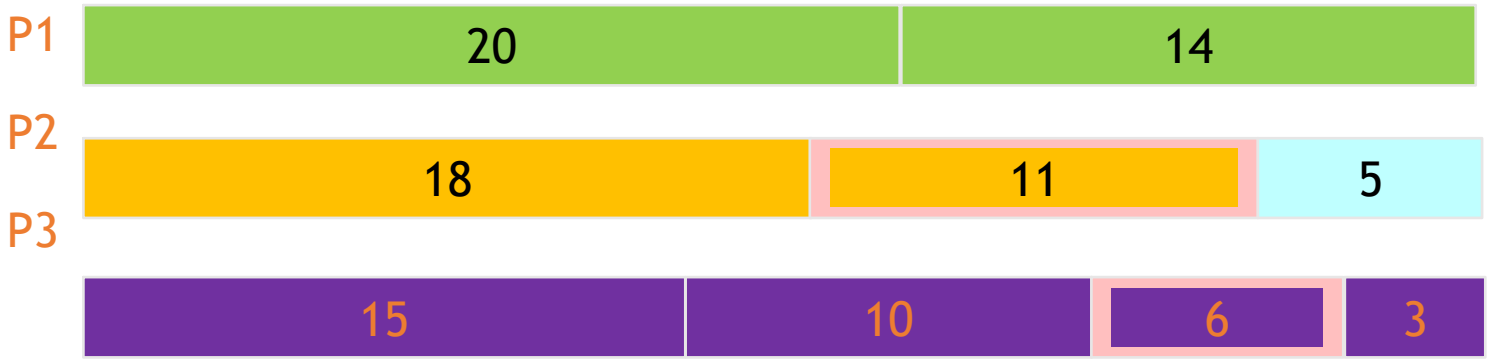
Jawaban:

- Jika dijalankan pada prosesor dengan job waktu terpendek lebih dahulu, hasilnya sbb:



Strategi ini bukan solusi yang bagus karena waktu penyelesaiannya menjadi 6 + 14 + 20 = 40 menit.

- Jika solusinya seperti ini: jalankan job dengan waktu terpanjang lebih dahulu pada setiap prosesor, solusinya lebih baik:

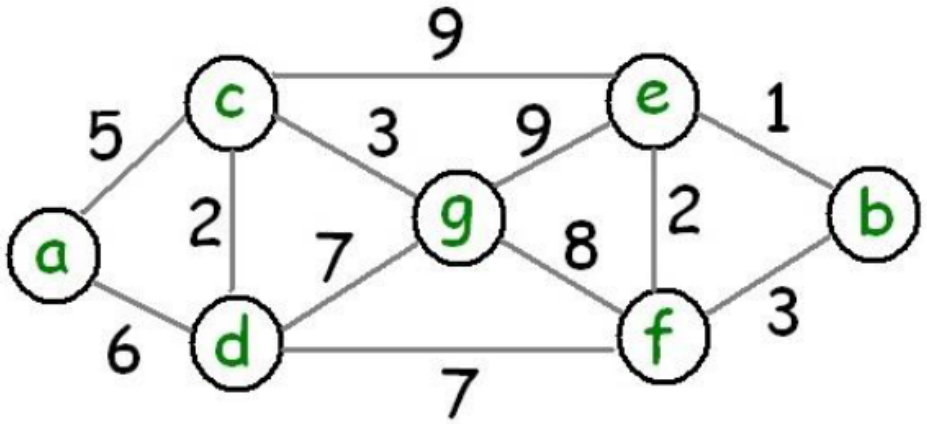


Waktu penyelesaiannya menjadi $20 + 14 = 34$ menit, optimal.

Apakah ada *counterexample?*

Soal Latihan 3

Diberikan graf berikut ini, kita akan menentukan lintasan terpendek dari simpul a ke semua simpul lainnya.



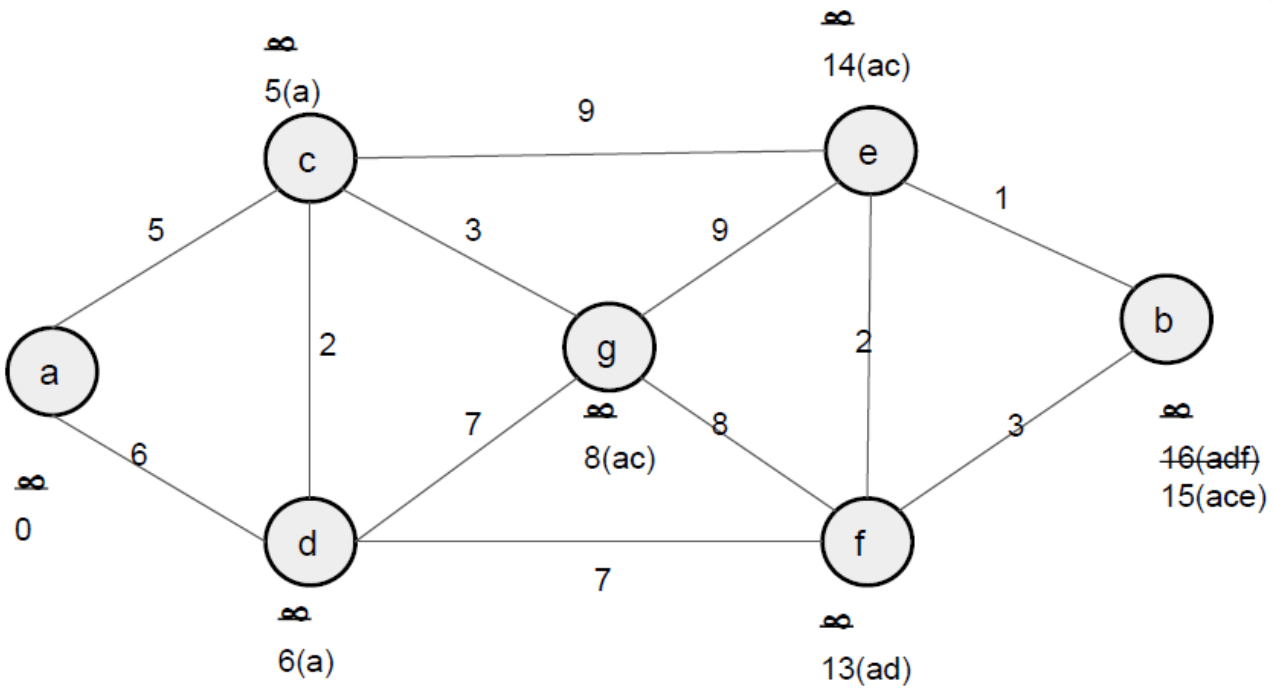
(a) Gunakanlah algoritma Kruskal untuk menentukan lintasan terpendek tersebut. Sebelum mengerjakan, tuliskanlah strategi greedy yang digunakan Kruskal. **(Nilai 7.5)**

(b) Gunakanlah algoritma Dijkstra untuk menentukan lintasan terpendek tersebut. Sebelum mengerjakan, tuliskanlah strategi greedy yang digunakan Dijkstra. **(Nilai 10)**

(c) Berikanlah kesimpulan dari hasil (a) dan (b). **(Nilai 2.5)**

Jawaban:

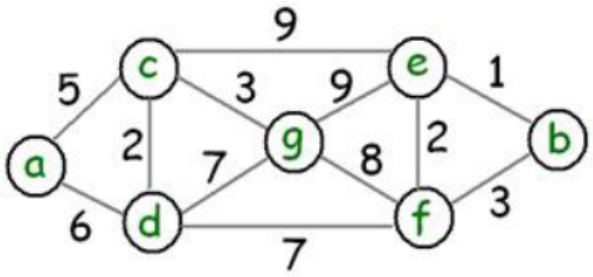
Solusi Soal a: Dijkstra (Nilai 10)



Lintasan terpendek Dijkstra:

a-b: a-c-e-b = 15; a-c: a-c = 5; a-d: a-d = 6

a-e: a-c-e = 14; a-f: a-d-f = 13; a-g: a-c-g = 8

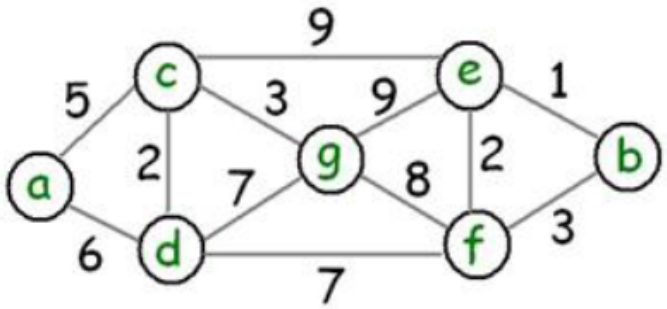


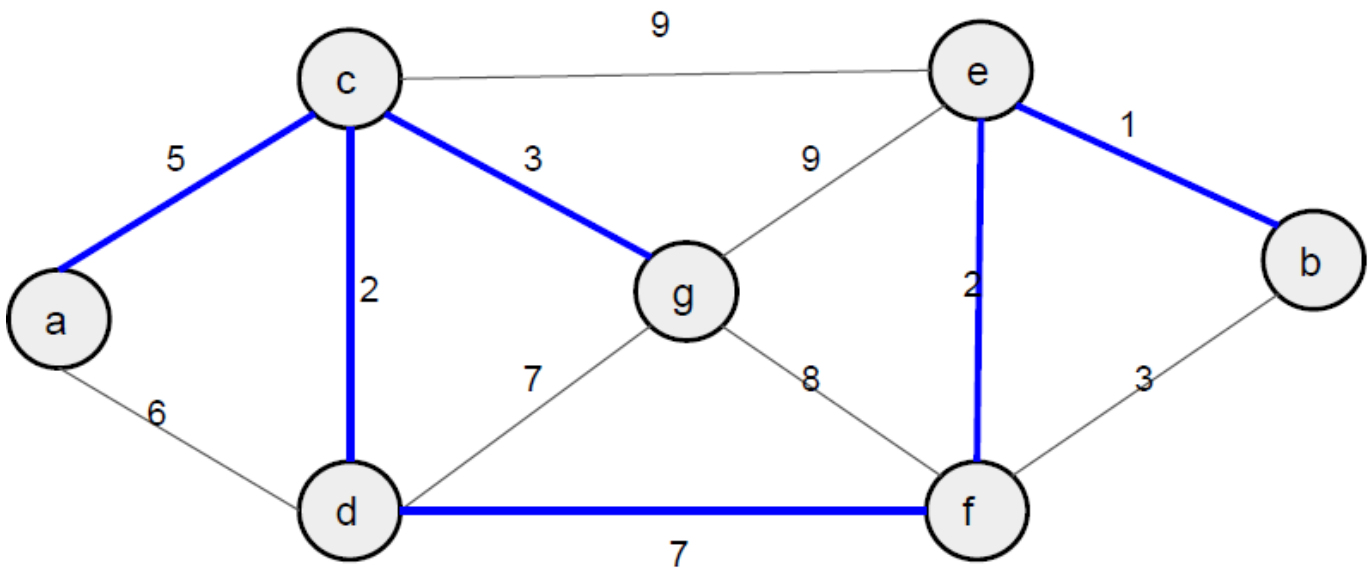
Strategi *greedy*

Pada setiap langkah, pilih simpul yang belum terpilih dan memiliki panjang lintasan terpendek dari simpul awal, lalu update simpul lain yang belum terpilih dengan L(u) + G(u,v) jika L(u)>L(u) + G(u,v).

(versi slide kuliah): Pada setiap langkah, ambil sisi yang berbobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan sebuah simpul lain yang belum terpilih.

Solusi b: Kruskal (nilai 7.5)





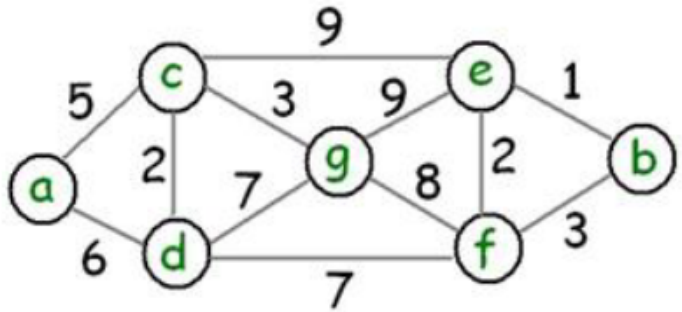
Strategi *greedy*:

Pada setiap langkah, pilih **sisi** *e* dari graf *G* yang mempunyai bobot minimum tetapi *e* tidak membentuk sirkuit di *T*.

Lintasan terpendek:

a-b: a-c-d-f-e-b = 17
a-c: a-c = 5
a-d: a-c-d = 7
A-e: a-c-d-f-e = 16
A-f: a-c-d-f = 14
A-g: a-c-g = 8

Solusi c (nilai 2.5)



Lintasan terpendek Dijkstra:
a-b: a-c-e-b = 15; a-c: a-c = 5; a-d: a-d = 6
a-e: a-c-e = 14; a-f: a-d-f = 13; a-g: a-c-g = 8

Lintasan terpendek Kruskal:
a-b: a-c-d-f-e-b = 17; a-c: a-c = 5; a-d: a-c-d = 7
a-e: a-c-d-f-e = 16; a-f: a-c-d-f = 14; a-g: a-c-g = 8

Kesimpulan:
Dijkstra menjamin memberikan lintasan terpendek ke semua simpul; sedangkan Kruskal hanya memberikan pohon merentang minimum yang tidak memberikan lintasan terpendek ke semua simpul.

Soal Latihan 4

- Setiap seminar promosi produk akan memberikan *cash-back* yang diasumsikan sama besarnya, sehingga setiap pelanggan berusaha mengikuti seminar promosi sebanyak-banyaknya. **Definisikanlah strategi greedy** yang dapat digunakan untuk menyelesaikan persoalan berikut, lalu **berikanlah solusinya**. Bandingkanlah **kompleksitas algoritmanya dengan strategi exhaustive search**. Misalkan pelanggan membeli 8 produk yang mengadakan seminar promosi dengan informasi sbb:

Produk	Waktu mulai	Waktu selesai
makanan beku 1	1	4
makanan beku 2	2	4
Elektronik 1	1	3
Elektronik 2	5	7
sayur & buah	4	7
susu 1	3	4
Susu 2	6	8
Mie 1	4	5
Mie 2	7	8

Jawaban Soal 1

- Strategi greedy:
  - Urut berdasarkan waktu selesai menaik
  - Pilih seminar yang waktu mulainya lebih besar atau sama dengan waktu selesai seminar yang dipilih sebelumnya

- Solusi:
  Elektronik1 (1,3), Susu 1 (3,4),
  mie1 (4,5), elektronik 2 (5,7), mie 2 (7,8)
- Kompleksitas greedy:
  $O(n \log n) + O(n)$
- Kompleksitas exhaustive search:
  $O(n.2^n)$

Produk	Waktu mulai	Waktu selesai	Durasi
Elektronik 1	1	3	2
Makanan beku 1	1	4	3
Makanan beku 2	2	4	2
Susu 1	3	4	1
Mie 1	4	5	3
Elektronik 2	5	7	1
Sayur dan buah	4	7	3
Susu 2	6	8	2
Mie 2	7	8	1

Algoritma Greedy: O(n log n) + O(n)

```
function ActivitySchedulling(input C:himpunan_act) → himpunan_act
{ Menghasilkan barisan activity yang akan dilakukan}
Deklarasi
  i : integer
  A : himpunan_act   { solusi }
Algoritma
  A ← {}
  sort C berdasarkan strategi greedy    //O(n^2)
  while C ≠ {} do          /iterasi dilakukan n kali
     i ← activity pertama pada C yang sudah terurut
     C ← C - {i}
     if (A ∪ {i} layak atau tidak bentrok) then
        A ← A ∪ {i}
     endif
  endwhile
  { C = {} }
  return A
```

Algoritma Greedy: $O(n^2)$

```
function ActivitySchedulling(input C:himpunan_act) → himpunan_act
{ Menghasilkan barisan activity yang akan dilakukan}
Deklarasi
  i : integer
  A : himpunan_act    { solusi }
Algoritma
  A ← {}
  while C ≠ {} do                    //iterasi dilakukan n kali
     i ← activity terbaik sesuai strategi greedy    //O(n)
     C ← C - {i}
     if (A ∪ {i} layak atau tidak bentrok) then
        A ← A ∪ {i}
     endif
  endwhile
  { C = {} }
  return A
```

Exhaustive Search: O(n.2^n)

```
function ActivitySchedulling(input C:himpunan_act) → himpunan_act
{ Menghasilkan barisan activity yang akan dilakukan}
Deklarasi
  i : integer
  A : himpunan_act   { solusi}
  SC: array of himpunan_act //kandidat solusi
  kinerja: array of number  //sesuai fungsi objektif
Algoritma
  SC ← generateAllSubset(C)
  Foreach A ∈ SC do                  //iterasi dilakukan 2^n kali
     if (A layak atau tidak bentrok) then  // O(n)
        kinerja[i]=fungsiObjektif(A)
     else kinerja[i]=null                  // tidak layak
     endif
  endwhile
  return elemen SC dengan kinerja tertinggi
```

Soal Latihan 5

Terdapat n buah mata kuliah yang akan dijadwalkan pada sejumlah ruang kuliah. Setiap mata kuliah i memiliki waktu mulai si dan waktu selesai fi. Bagaimana menjadwalkan semua kuliah pada ruang-ruang kuliah sehingga jumlah ruang kuliah yang dipakai seminimal mungkin? Tidak boleh ada dua atau lebih kuliah yang bentrok waktunya (beririsan waktunya) menggunakan ruang kuliah yang sama. Jelaskan strategi greedy-nya seperti apa dan berapa kompleksitas waktunya. Ilustrasikan jawaban anda dengan contoh berikut: (Nilai: 15)

n = 10, (si , fi) = [(5, 6), (4, 7), (2, 5), (1, 4), (3, 7), (8, 10), (7, 8), (1, 3), (6, 9), (5, 8)]

Jawaban:
Strategi *greedy*-nya adalah:

(i) Urutkan mata kuliah dalam urutan menaik berdasarkan waktu mulainya (s).

(ii) Mulai dengan ruang kuliah ke-k = 1.

(iii)Assign kuliah-kuliah ke dalam ruang kuliah k yang waktu mulainya lebih besar atau sama dengan waktu selesai kuliah yang telah dipilih sebelumnya.

(iv)Jika ada kuliah yang tersisa, tambahkan ruang kuliah baru (k = k+ 1), lalu ulangi langkah (iii) sampai seluruh mata kuliah sudah di-*assign* ke ruang-ruang kuliah.

Contoh: n = 10, (s_i , f_i) = [(5, 6), (4, 7), (2, 5), (1, 4), (3, 7), (8, 10), (7, 8), (1, 3), (6, 9), (5, 8)]

Diurutkan: [(1, 3), (1, 4), (2, 5), (3, 7), (4, 7), (5, 6), (5, 8), (6, 9), (7, 8), (8, 10)]

Ruang kuliah k = 1: [(1, 3), (3, 7), (7, 8), (8, 10)]

Ruang kuliah k = 2: [(1, 4), (4, 7)]

Ruang kuliah k = 3: [(2, 5), (5, 6), (6, 9)]

Ruang kuliah k = 4: [(5, 8)]

Jadi, dibutuhkan 4 ruang kuliah saja

Meng-*assign* setiap kuliah ke ruang kuliah cukup dilakukan dalam waktu O(n) saja. Kompleksitas algoritma adalah $O(n^2)$ atau O(n log n) jika waktu pengurutan diperhitungkan.

Jika waktu pengurutan tidak diperhitungkan, maka kompleksitasnya adalah O(n).

Soal Latihan 6

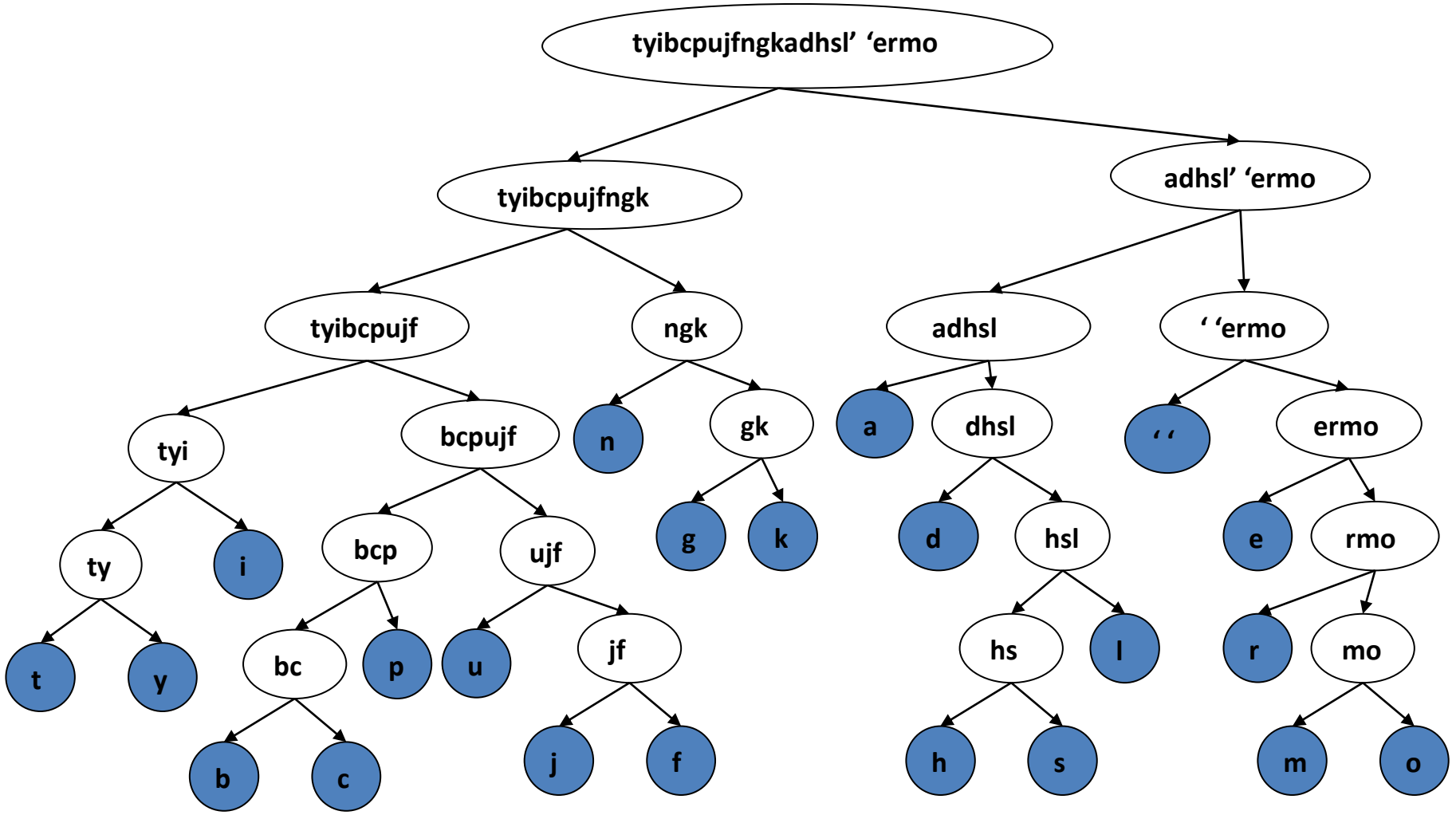
- Lakukan encoding dengan algoritma Huffman untuk teks berikut ini: “prinsip kode huffman karakter yang paling sering muncul di dalam data dengan kode yang lebih pendek sedangkan karakter yang relatif jarang muncul dikodekan dengan kode yang lebih panjang”.

  Untuk mempermudah, berikut adalah frekuensi setiap karakter:

  { =27, a=24, b=2, c=2, d=12, e=15, f=3, g=11, h=3, i=9, j=2, k=11, l=7, m=4, n=21, o=4, p=5, r=8, s=3, t=4, u=5, y=4}

  Jika terdapat karakter dengan frekuensi yang sama, karakter yang lebih kecil (a<b<c<...<z) berada di sebelah kiri karakter lainnya. Pohon gabungan terbaru selalu diletakkan di sebelah kanan dari pohon lain yang memiliki frekuensi yang sama.

Pohon Huffman



Pembentukan pohon

{‘ ’=27, a=24, b=2, c=2, d=12, e=15, f=3, g=11, h=3, i=9, j=2, k=11, l=7, m=4, n=21, o=4, p=5, r=8, s=3, t=4, u=5, y=4}

Urut berdasarkan frekuensi kecil ke besar:

b=2, c=2, j=2, f=3, h=3, s=3,m=4, o=4, t=4, y=4, p=5, u=5, l=7, r=8, i=9, g=11, k=11, d=12, e=15, n=21, a=24, ‘ ’=27

j=2, f=3, h=3, s=3,m=4, o=4, t=4, y=4, bc=4,p=5, u=5, l=7, r=8, i=9, g=11, k=11, d=12, e=15, n=21, a=24, ‘ ’=27

h=3, s=3,m=4, o=4, t=4, y=4, bc=4,p=5, u=5, jf =5, l=7, r=8, i=9, g=11, k=11, d=12, e=15, n=21, a=24, ‘ ’=27

m=4, o=4, t=4, y=4, bc=4,p=5, u=5, jf=5,hs=6,l=7, r=8, i=9, g=11, k=11, d=12, e=15, n=21, a=24, ‘ ’=27

t=4, y=4, bc=4,p=5, u=5, jf=5,hs=6,l=7, r=8, mo=8, i=9, g=11, k=11, d=12, e=15, n=21, a=24, ‘ ’=27

bc=4,p=5, u=5, jf=5,hs=6,l=7, r=8, mo=8, ty=8, i=9, g=11, k=11, d=12, e=15, n=21, a=24, ‘ ’=27

u=5, jf=5,hs=6,l=7, r=8, mo=8, ty=8, i=9, bcp=9, g=11, k=11, d=12, e=15, n=21, a=24, ‘ ’=27

hs=6,l=7, r=8, mo=8, ty=8, i=9, bcp=9, ujf=10, g=11, k=11, d=12, e=15, n=21, a=24, ‘ ’=27

r=8, mo=8, ty=8, i=9, bcp=9, ujf=10, g=11, k=11, d=12, hsl=13, e=15, n=21, a=24, ‘ ’=27

ty=8, i=9, bcp=9, ujf=10, g=11, k=11, d=12, hsl=13, e=15, rmo=16, n=21, a=24, ‘ ’=27

bcp=9, ujf=10, g=11, k=11, d=12, hsl=13, e=15, rmo=16, tyi=17, n=21, a=24, ‘ ’=27

g=11, k=11, d=12, hsl=13, e=15, rmo=16, tyi=17, bcpujf=19, n=21, a=24, ‘ ’=27

d=12, hsl=13, e=15, rmo=16, tyi=17, bcpujf=19, n=21, gk=22, a=24, ‘ ’=27

e=15, rmo=16, tyi=17, bcpujf=19, n=21, gk=22, a=24, dhsl=25, ‘ ’=27

tyi=17, bcpujf=19, n=21, gk=22, a=24, dhsl=25, ‘ ’=27, ermo=31,

n=21, gk=22, a=24, dhsl=25, ‘ ’=27, ermo=31, tyibcpujf=36,

a=24, dhsl=25,‘ ’=27, ermo=31, tyibcpujf=36, ngk=43,

‘ ’=27, ermo=31, tyibcpujf=36, ngk=43, adhsl=49,

tyibcpujf=36, ngk=43, adhsl=49, ‘ ’ermo=58,

adhsl=49, ‘ ’ermo=58, tyibcpujfngk=79,

tyibcpujfngk=79, adhsl‘ ’ermo=107,

tyibcpujfngkadhsl‘ ’ermo=186,

((((t-y)-i)-(((b-c)-p)-(u-(j-f))))-(n-(g-k)))-((a-(d-((h-s)-l)))-(‘ ’-(e-(r-(m-o)))))