



Tisdagen den 18 mars 2014 kl 9–13

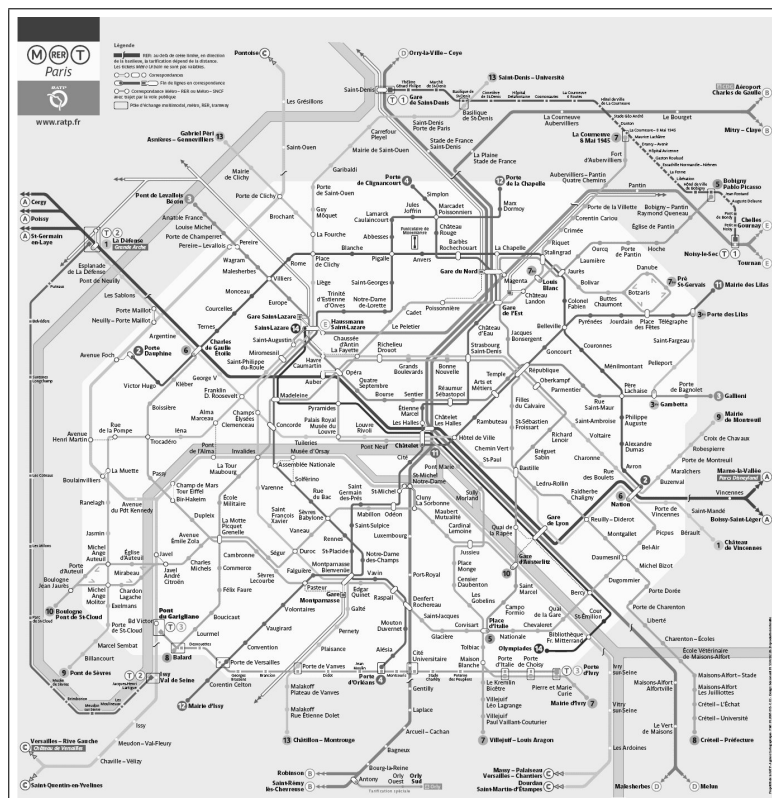
Hjälpmedel: En algoritmbok och ditt eget formelblad. För betyg E krävs att alla E-uppgifter är godkända, för betyg C att alla E- och C-uppgifter är godkända, för betyg A att alla uppgifter är godkända.

1. Var är bussen?

Betyg E. Om en buss är inställd men bussentreprenören ”glömmer bort” att rapportera detta så får det märkliga effekter på den elektroniska anslagstavlan vid busshållplatser. Den kommer att gå på tidtabellen när ingen kontakt fås med bussen och då kanske man får se följande nedräkning:
 4, 3, 2, 1, Nu, 15, 14
 Trots att det står ”Nu” kommer ingen buss...
 För att söka efter missade bussar i en text kan man använda Boyer-Moore. Visa hur Boyer-Moore fungerar när man söker efter ”178” i följande text:
 ”Lätt snöfall gör så att buss 70, 178 och 677 är kraftigt försenade”
 Det är OK att hoppa över steg i mitten men var mycket noga med de sista jämförelserna. (15 min)

2. Resa snabbt med Metron

Betyg E. Paris pendel- och tunnelbana är ett virrvarr av olika stationer som står i förbindelse med varandra. Beskriv översiktligt en effektiv algoritm som räknar ut en resväg från en station till en annan med så få stationsstopp som möjligt. Att byta tåg på en station räknas som samma station. (15 min)



3. *Tidsordning i binärt sökträd*

Betyg E. Tunnelbanetiderna för en station är insorterade i ett balanserat binärt sökträd. Beskriv en rekursiv algoritm som skriver ut tiderna i ordning:
6:15 6:45 7:00 7:15 7:30
o.s.v. (10 min)

4. *Skriv ut hållplatstabell med binärt sökträd*

Betyg C. I tunnelbanan är tiderna utskrivna med timmen längst till vänster och avgångsminuterna till höger. Beskriv en algoritm som skriver ut tunnelbanetiderna från det binära sökträdet på följande sätt:
6 15 45
7 00 15 30 45
8 00 15 ...
. .
(15 min)

5. *Nästa avgångstid i binärt sökträd*

Betyg A. Beskriv en effektiv algoritm som givet en tid söker efter nästa avgångstid i det binära sökträdet med tunnelbanetider. Var noga med de sista jämförelserna i algoritmen.
Algoritmbeskrivningen måste vara korrekt, välstrukturerad och begriplig för betyg A. (25 min)

6. *Stoppa in i heap*

Betyg E. Givet en heap
[200, 128, 5, 81, 112, 3]
Vilket av dessa alternativ hur heapen ser ut efter insättning av 7 och därefter 9 i heapen:
a) [200, 128, 112, 81, 7, 3, 5, 9]
b) [200, 128, 7, 81, 112, 3, 5, 9]
c) [200, 128, 81, 9, 112, 3, 7, 9]
d) [81, 7, 128, 5, 9, 112, 200, 3]
e) [200, 128, 9, 81, 112, 3, 5, 7]
f) [200, 128, 9, 81, 112, 5, 3, 7]
(15 min)

7. *Heap eller inte heap?*

Betyg E. Är det något eller några av alternativen ovan som inte kan vara en heap? Motivera kort. (10 min)

8. *Resonera kring syntax*

Betyg C. Anta att du har skrivit en syntax för kommunikation mellan lokförare och trafikledning.

Beskriv två olika sorters testfall som man kan testa syntaxen med.

(15 min)

9. *Bygg Huffmanträd*

Betyg E. Rita ett huffmanträd givet frekvenserna nedan.

A 50%

N 33.3%

S 16.7%

(10 min)

10. *Avkoda Huffmanträd*

Betyg A. Man har ett huffmankodat meddelande, det är 010010011.

Går det alltid att avkoda något som är huffmankodat om man bara har frekvenstabellen och inte huffmanträdet? Motivera ditt svar. *(20 min)*

11. *Datastruktur för resdata*

Betyg C. SL samlar data om dina resor. Beskriv en eller flera datastrukturer som lagrar information om var och när ett SL-kort har använts i SL-trafiken.

Det ska gå snabbt att slå upp informationen givet SL-kortets id. *(15 min)*

12. *Sortera undanflykter*

Betyg C. En SL-resenär har en sorterad lista (i alfabetisk ordning) över de undanflykter som SL:s lokala stabschef förklarar förseningarna med. Problemet är att dessa små lögner ynglar av sig så att det dyker upp en eller två nya osorterade lögner i slutet på listan.

Visa hur man med bubbelsortering effektivt kan sortera denna lista igen. *(15 min)*