Avsluttende eksamen i TDT4125
Algoritmekonstruksjon, videregående kurs (Bokmål)

Kontakt under eksamen	Magnus Lie Hetland (tlf. 91851949)
Tillatte hjelpemidler	Alle trykte/håndskrevne; bestemt, enkel kalkulator

Bruk gjerne blyant! Les hele oppgavesettet først, disponer tiden og forbered evt. spørsmål til faglærer kommer til eksamenslokalet. Gjør antagelser der det er nødvendig. Svar konsist, fortrinnsvis i svarskjemaet.

Svarskjema

1a (9%)
1b (9%)
1c (9%)
2a (9%)
2b (9%)

3a (9%)

3b (9%)

4a (9%)

4b (9%)

5a (9%)

5b (10%)

Oppgave 1

a. Du ønsker å avgjøre hvorvidt to logiske formler kan ha samme verdi under de samme omstendighetene (dvs., evt. felles variable har samme verdi). Vis kort at dette er et NP-komplett problem.

b. Du ønsker nå å avgjøre hvorvidt to logiske formler alltid vil ha samme sannhetsverdi under samme omstendigheter (samme verdier for felles variable). Vis kort at dette er et co-NP-komplett problem.

Følgende (formodentlig gale) utsagn er tatt fra Wikipedia:[1]

> A problem is said to be strongly NP-hard if a strongly NP-complete problem has a polynomial reduction to it.[2]

(Merk at sterk NP-komplettheet er et spesialtilfelle av NP-komplettheet.)

c. Vis at utsagnet implisitt besvarer spørsmålet om hvorvidt P = NP.

Oppgave 2

a. Beskriv hvordan Floyd-Warshall kan tilpasses en PRAM-maskin med n^2 prosessorer slik at du får en kjøretid på $\Theta(n)$, der n er antall noder. Hvilken PRAM-modell trenger du?

(Du kan gjenbruke/skrive over avstandsmatrisen, så du unngår kubisk allokeringstid.)

Kjernen i Floyd-Warshall kan uttrykkes rekursivt som følger:

D(u, v, k):
if $k = 0$
    return $W[u, v]$
return min { D(u, v, k–1), D(u, k, k–1) + D(k, v, k–1) }

b. Hvis du antar at hvert rekursive kall utføres i parallell (med **spawn**), hva er parallellitetsgraden (*parallelism*) til prosedyren D, som funksjon av k? Vis kort utregning.

Oppgave 3

Anta at du har et sett med n punkter i planet. Hvert punkt har én av k farger (og alle fargene er representert). Du ønsker å velge ut k punkter, ett av hver farge, slik at omkretsen (perimeteren) av det konvekse skallet (*convex hull*) rundt de k punktene blir så liten som mulig.

Betrakt følgende grådige approksimeringsalgoritme for dette problemet: Velg et punkt p, og for hver farge c forskjellig fra p sin, velg et punkt av farge c som er nærmest p. Gjenta denne prosedyren for ethvert punkt p, og velg til slutt den av de n punktmengdene som har et konvekst skall med minst diameter. (Diameteren er avstanden mellom de to punktene som er lengst fra hverandre.)

a. Vis at denne algoritmen har en approksimerings-ratio (*approximation ratio*) på $\pi/2$.

Hint: Anta at det finnes en gyldig løsning med diameter d. Hva kan du da si om diameteren til den optimale løsningen? Hva med den grådige?

---

[1] http://en.wikipedia.org/wiki/Strongly_NP-complete

[2] På norsk: «Et problem sies å være sterkt NP-hardt dersom et sterkt NP-komplett problem har en polynomisk reduksjon til det.»

Betrakt nå en billigere variant av den grådige algoritmen. I stedet for å velge løsningen med minst diameter, velg løsningen med minst omkrets.

b. Vis at denne algoritmen har en approksimerings-ratio på π.

Hint: Anta at den optimale løsningen har diameter *d*. Da finnes det et punkt *p* som har punkter av alle de andre fargene innen en avstand på *d*. Dette punktet vil naturligvis vurderes av den grådige algoritmen.

Oppgave 4

I *bin-packing*-problemet har du en mengde objekter av ulike størrelser som skal plasseres i så få båser (*bins*) som mulig, der båsene har fast størrelse. Mer spesifikt, gitt en mengde av elementer med størrelser $S = \{s_1, \dots, s_n\}$ og en bås-størrelse L ($L \geq s_i$ for alle s_i i S) er målet å finne en B-partisjonen $S_1 \cup \dots \cup S_B$ av $\{1, \dots, n\}$ som er slik at $\sum_{i \in Sk} s_i \leq L$ for alle $k = 1, \dots, B$ and slik at B er minimal. For eksempel, for $S = \{1, 3, 4, 2, 2\}$ og $L = 6$ vil en slik optimal B-partition være $\{1, 2, 4\} \cup \{3, 5\}$; det vil si, vi plasserer 1,3, and 2 i én bås og 4 and 2 i en annen bås.

a. Gitt S = {1, 3, 4} og $L = 4$, tegn et søketre som representerer de mulige løsningene til *bin-packing*-problemet. Hvor mange optimale løsninger inneholder søketreet ditt?

b. Anta at du har en partiell løsning P, der et element s_i har blitt plassert i en bås. Beskriv en algoritme for å beregne et underestimat $g(P)$ for den beste mulige løsningen du kan få (antall båser), gitt din partielle løsning P. Hva er kjøretiden på algoritmen din?

Oppgave 5

Du skal planlegge bruken av piloter for et flyselskap. Du har oppgitt en liste med byer, tiden det tar å fly fra hver by til hver av de andre (der det går fly) og en liste med flyvninger. For hver flyvning vet du byen den starter i, byen den lander i, samt tiden den starter. Du har *n* piloter som skal fordeles på flyvningene.

a. Beskriv en effektiv algoritme som løser problemet. (Bruk gjerne en figur.)

En flyt kan inneholde sykler – dvs. vi kan ha en sykel i flytnettverket der alle kantene har tilordnet flyt større enn 0. I mange anvendelser vil dette representere bortkastet arbeid, der vi transporterer en ressurs i ring, heller enn å få den dit den skal. En *effektiv* flyt er en flyt uten sykler.

b. Skisser en algoritme for å finne en effektiv maksimal flyt for et gitt flytnettverk. Forklar kort hvorfor algoritmen er korrekt.