1.

(Odgovor: b0ysee) Za povezani graf G kazemo da je eulerovski ako postoji zatvorena staza koja sadrzi svaki brid od G. Povezani graf G je eulerovski onda i samo onda ako je stupanj svakog vrha paran.
Iz ovih tvrdnji moze se za Kr,s reci kako i r i s moraju biti parni posto je Kr,s potpuni bipartitan graf (svaki vrh s jedne strane vezat ce se na sve vrhove s druge, a njegov ce stupanj biti paran samo ako se s druge strane nalazi paran broj vrhova). Takodjer, graf moze biti i skoro eulerovski, a skoro eulerovski je ako staza kroz sve bridove nije zatvorena, te vrijedi da je graf skoro eulerovski ako i samo ako ima dva vrha s neparnim stupnjem. To znaci da ili r ili s mogu biti 2, a onda se s druge strane mora nalaziti neparan broj vrhova (jer tada imamo 2 vrha koja su neparni, zbog druge strane, te sve ostale vrhove koji se vezu na ova dva i time su automatski parni).
Sto se kotaca Wn tice, on nikada nece biti niti eulerovski niti skoro eulerovski. To se najlakse vidi iz toga ako mu pogledamo "vanjske" vrhove. Oni uvijek imaju stupanj 3 posto imaju 2 susjedna "vanjska" vrha i 1 "unutarnji", tako da ce (n-1) vrhova (tj. ovi "vanjski") tog kotaca sigurno biti neparno, a tu vec pada test na eulerovski graf.

2.

(Odgovor: Pylo) Za povezani graf kažemo da je eulerovski graf, ako postoj zatvorena staza koja sadrži svaki njgov brid. U eulerovskom grafu svi vrhovi imaju parni stupanj.

Ako neki vrh ima paran stupanj, znači da je incidentan sa parnim brojem bridova. To znači da će svaki brid u tom vrhu imati neparan broj susjednih bridova, ali kako svaki brid spaja dva vrha, i u svakom ima neparan broj susjednih bridova, ukupno će imati paran broj susjednih (zbroj dva neparna broja = paran).
Bridni graf preslikava te bridove u vrhove, koji će tada također imati jednak broj susjednih vrhova, dakle paran broj, a to je ujedno i stupanj nekog vrha.
Zato je bridni graf jednostavnog eulerovskog grafa također eulerovski.

3.

(Odgovor: Pylo) Ovo je jako jednostavno i poprilično dosadno.
Fleuryev aloritam je postupak pronalaženja eulerovske staze (ako postoji), a radi se na slijedeći način:
Počmete u bilo kojem vrhu grafa i "hodate" po grafu bilo kojim redoslijedom uz ova pravila:
1) prebriši bridove kojima si prošao, a ako vrh ostane izoliran, briši i njega
2) prijeđi mostom samo ako nemaš druge mogučnosti.

Sad je jedino problem nacrtati Q4 (to je hiperkocka za sve vas geekove:), ali naravno mi se nećemo time mučiti nego ćemo pogledati na internet.
http://en.wikipedia.org/wiki/Tesseract
Tu sam odabrao graf koji mi se najviše svidio i nasumice označio vrhove (nije važno kako ih označimo), i to onda ovako izgleda:

Postupak sam već objasnio, on nije jednoznačan i svatko može ići drugačijim putem. Ovako sam ja išao:

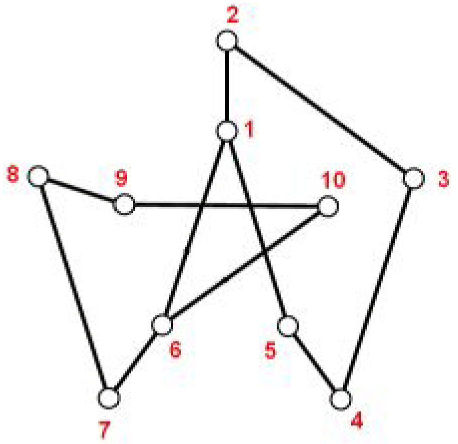
Eulerovska staza je: 1->2->3->4->5->6->7->8->1->10->3->12->5->14->7->16->11->14->9->12->15->10->13->6->15->8->9->2->11->4->13->16->1

4.

(Odgovor: Pylo) Petersenov graf imate u knjizi na strani 23, ali kome se ne da otvarat može ga vidjet ovdje: http://en.wikipedia.org/wiki/Petersen_Graph

Možete vidjet da taj graf ima 10 vrhova stupnja 3. Da bi graf bio skoro eulerovski on mora imati samo dva vrha neparog stupnja. Znači moramo od 10 vrhova, 8 smanjih stupanj, a svakim bridom koji maknemo utječemo na dva vrha. Dakle treba maknuti 4 nesusjedna brida, i to se može na ovom grafu.

Jedan primjer bi bio:



A staza je: 1->2->3->4->5->1->6->7->8->9->10->6

Primjetite da kod skoro eulerovskih grafova postoji staza koja prolazi kroz sve bridove, ali ona nije zatvorena. Ta staza počinje u jednom vrhu neparnog stupnja, a završava u drugom.

5.

(Odgovor: b0ysee) Ovdje nije potrebno crtati da bi se ustanovilo koliko bridova treba dodati. Naime, treba pogledati sve vrhove Petersonovog grafa.

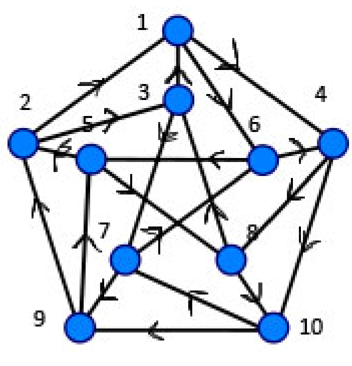
Svi njegovi vrhovi su stupnja 3, a mi zelimo dodavanjem bridova sve njih dovesti na stupanj 4 (tj. paran stupanj) kako bi dobili eulerovski graf.

Sve skupa postoji 10 vrhova u Petersonovom grafu. Jedan brid povezuje dva vrha i upravo ce novonastali brid povecati stupanj tocno tom broju vrhova.

Kako bi svim vrhovima podigli stupnjeve dovoljno je dodati 5 bridova (od kojih svaki poduze stupanj dvama vrhovima).

Sto se eulerovske staze tice, dobro ce doci crtez cisto da znamo kuda idemo.

Ovdje je najpametnije primijeniti Fleuryev algoritam koji je stvarno lagan.



Krenuo sam iz 1 prema cetvorci i nakon toga samo pratite strelice, a evo ovdje i eulerovska staza:

1 -> 4 -> 10 -> 9 -> 2 -> 1 -> 6 -> 4 -> 8 -> 10 -> 7 -> 9 -> 5 -> 3 -> 7 -> 6 -> 5 -> 8 -> 3 -> 1

6.

(Odgovor: Pylo) Graf vam je na slici (str. 36.) treba dokazati da je hamiltonovski. Hamilonovski grafovi imaju ciklus koji prolazi svim vrhovima (samo jednom).
U knjizi piše:
Ako je G jednostavni graf s n vrhov, n>=3, te ako vrijedi:
deg(v)+deg(w)>=n
za svaki par nesusjednih vrhova v i w grafa G, onda je G hamiltonovski.

Drugim riječima mi bi trebali uzet svaka dva nesusjedna vrha u grafu, zbrojit njihove stupnjeve i provjerit da li je taj zbroj >= n (n=11 u grafu iz zadatka)
To je nekih 11! računica, pa neće biti problem. NOT! 

Ali već ćemo na prvom pokušaju zbrajanja stupnjeva nesusjednih vrhova dobiti manje od 11, pa očito ova pretpostavka ne pomaže. Što nam preostaje? Da baš to, crtanje. 

Evo primjera jednog takvog ciklusa. Ide od 1 do 11, pa se vrati na 1.

7.

(Odgovor: b0ysee) Takvi grafovi su upravo ciklus grafovi - Cn. Da bi graf imao eulerovsku stazu mora postojati zatvorena staza koja prolazi kroz sve bridove - to vrijedi za Cn. Da bi graf bio hamiltonov, tj. imao hamiltonovu stazu mora imati zatvorenu stazu koja prolazi kroz sve vrhove, a i to Cn ocito ima.

8.

(Odgovor: Pylo) Potpuni graf K(2k+1) ima [(2k+1)(2k)/2] = k(2k+1) bridova. Budući da svaki disjunktni hamiltonovski ciklus zauzima (2k+1) bridova (jer prolazi kroz sve vrhove), može ih biti najviše:
k(2k+1)/(2k+1) = k

9.

(Odgovor: Pylo) Pretpostavimo da imamo potpuni graf sa n vrhova, n>=3. On ima n(n-1)/2 bridova i stupanj svakog vrha mu je (n-1).
Zanima nas koliko mu bridova trebamo maknuti, da bi dobili graf sa (n^2-3n+6)/2 koji je zadan u zadatku.

[n(n-1)/2] - [(n^2-3n+6)/2] = n-3

Znači početnom potpunom grafu maknemo (n-3) brida i dobijemo graf sa zadanim svojstvima.

Sada pogledajmo da li takav graf zadovoljava dovoljan uvijet za sve nesusjedne vrhove v i w:
deg(v)+deg(w)>=n

U potpunom grafu taj će zbroj uvijek biti 2(n-1) što je veće od n. Ako maknemo (n-3) bridova, u najgorem slučaju nekom će se vrhu stupanj smanjit za (n-3).
Treba predvidjeti još jednu mogućnost, a to je da je jedan od bridova koje smo maknuli spajao oba promatrana vrha, pa je zato i drugom vrhu stupanj pao za 1.
Dakle od dva promatrana nesusjedna vrha moguće je da im se zbroj stupnjeva smanjio najviše za [(n-3)+1]=n-2
Zbroj stupnjeva će tada biti:
2(n-1) - (n-2) = n >=n

Uvijet je zadovoljen, a pretpostavka dokazana

10.

Zadatak nas trazi broj bridova koji se smije oduzeti svakom potpunom grafu (n>=4), ali tako da on i dalje ostane hamiltonovski.

Teorem 13 na strani 33 kaze ovako:
Ako je G jednostavni graf s n (n>=3) vrhova, te ako je deg(v) >= n/2 za svaki vrh iz G, tada je G hamiltonovski.

Mi znamo da svaki vrh u potpunom grafu ima stupanj (n-1) posto je ima brid sa svim ostalim vrhovima (osim, naravno, sa samim sobom).
Ako na to tako gledamo, moramo samo ostvariti slucaj da graf promijenimo tako da svaki vrh ima stupanj tocno n/2.
To znaci da svakom vrhu mozemo smanjiti stupanj za (n/2 - 1) - ovaj -1 dolazi od cinjenice da je broj susjednih vrhova (n-1), konkretno:

n - 1 - (n/2 - 1) = n - 1 - n/2 + 1 = n - n/2 = n/2,

a to je upravo ono sto trazimo.
No, treba pripaziti jer rjesenje nije jednostavno (n/2 - 1), vec (n/2 - 1)/2, posto mi u ovom nasem slucaju gledamo vrhove i tako svaki brid brojimo dva puta.

11.

(Odgovor: b0ysee) Ovakvi zadaci najlakse se rijesavaju Dijsktrinim algoritmom s 41. str). Ukratko, ideja je da se usporedjuju susjedni vrhovi i bira najkraci put (s time da se uvodi varijabla l koja sadrzi ukupni najkraci put do trenutnog vrha), ali tako da se usporedjuje i sa korakom "unazad" (da se slucajno ne bi nasao u konacnici mozda povoljniji put). Na 42. str. ima lako pratljiv postupak, pa necu puno ulaziti u to, vec samo napisati po uzoru na taj primjer kako rjesiti ovaj zadatak.

l(A) = 0, l(B) = inf, l(C) = inf, ..., l(G) = inf, S0 = A, i = 0 (pocetno stanje)

l(B) = 30, l(C) = 50, min{30,50,inf} = 30, u1 = B, S1 = {A,B}, i = 1 (izabrali smo najkraci put do vrha B).

l(C) = 49, l(D) = 36, l(E) = 70, min{50, 49, 36, 70, inf} = 36, u2 = D, S2 = {A,B,D}, i = 2 (sljedeci vrh je vrh D posto on daje najkraci put, logicno)

l(C) = 48, l(F) = 59, l(E) = 71, min{49, 70, 48, 59, 71, inf} = 48, u3 = C, S3 = {A,B,D,C}, i = 3 (ovo je zanimljivo, naime, dosli smo u vrh C preko dva medjuvrha umjesto da smo isli direktno iz A, ali zato bas Dijkstrin algoritam ovako odlicno pali jer smo medjuputevima do C-a dosli sa "samo" 48 jedinica, a ovako bi nam trebalo 50. Takodjer, treba vidjeti da smo uzeli sve podatke iz proslog koraka kad smo trazili min kako ne bismo izgubili kakvo mozebitno rjesenje).

l(F) = 58, min{59, 58, inf} = 58, u4 = F, S4 = {A,B,D,C,F}, i = 4 (opet smo "ustedjeli" jednu jedinicu! :) Osim toga, usporedjivanje s E-om iz proslog koraka svodi se na usporedbu s inf. jer E nije susjedan vrh)

l(E) = 69, l(G) = 88, min{71, 69, 88} = 69, S5 = {A,B,D,C,F,E}, i = 5 (ovdje smo ipak morali usporediti s onim prijasnjim E-om jer je opet moglo doci do gubljenja rjesenja)

l(G) = 77, min{88, 77} = 77, S6 = {A,B,D,C,F,E,G}, i = 6, KRAJ!
(Dosli smo do G-a, naravno usporedjivali smo i s onom drugom varijantom da ne izgubimo koje rjesenje)

12.

(Odgovor: b0ysee) Ovaj zadatak moze se naravno rijesiti i Dijkstrinim algoritmom (pogotovo za udaljenije tocke), ali ne vjerujem da ce mi se dati, pa cu samo proci kroz vrhove gledajuci sliku jer se vecina vidi na prvi pogled. :)

* A

1 je najmanja tezina brida, pa ce to biti direktan put.

S -> A

* B

Moze se doci i preko A i preko C, ali nijedan takav put ne nudi tezinu manju od 3, tako da ce i to biti direktno.

S -> B

* C

Moze se doci preko A, no ta ruta bilo kako bilo nudi vrlo brzo put veci od 6, medjutim, preko vrha B moze se doci za 5 jedinica, tako da biramo taj put!

S -> B -> C

* D

Ovdje stvarno ne treba puno mozgati da bi se vidjelo kako se preko vrha A dodje za 3 jedinice do vrha D, tako da je to dosta ocito.

S -> A -> D

* E

E, ovdje vec ima malo vise mogucnosti, medjutim, put preko A i D nudi put od 4 jedinice. Ako krenemo preko B u startu imamo bar 3 jedinice (koje ce sigurno dati bar 4 kasnije), a ako krenemo preko C u startu imamo 6 jedinica, tako da je i tu odluka vrlo laka.

S -> A -> D -> E

* F

Put preko tocke A u najboljem slucaju iznosi 9 jedinica (samo 2 moguca puta koja su kolko tolko "stedljiva"). Put preko tocke C je nuzno veci od 9, pa nas ni ne zanima, ali preko tocke B mozemo stici s 3+3 jedinice, pa cemo to i izabrati! :)

S -> B -> F

* G

Kod G-a treba upotrijebiti do sada dobivene podatke i pogledati sto je najpametnije iskoristiti. Do vrha D dosli smo preko A najbrze za 3 jedinice. Ako skratimo put preko E dodat cemo mu samo 1+5 jedinica i on ce iznositi sve skupa 9. Do vrha E smo dosli najbrze preko E-a, no on je ukljucen u put od D (ovaj prvi promatrani), tako da nas ne zanima vise. Do vrha F smo dosli za 6 jedinica (preko tocke B), no on ima put koji nosi samo 2 jedinice, tako da moze ukupno doci za 8 jedinica do vrha G i ociti je pobjednik! :)

S -> B -> F -> G

Ovaj zadnji primjer (vrh G) zapravo je mini primjena Dijkstrinog algoritma bez pisanja svih onih varijabli.

13.

(Odgovor: b0ysee) Za ovaj zadatak nisam u potpunosti siguran, ali ja bio napravio sljedece modifikacije (koje su oznacene bold tekstom):

(algoritam s 41. strane)

1. Stavi l(u0) = 0, l... (ovdje ne bih nista mijenjao)
2. Za svaki vrh v E kompl(Si), zamijeni l(v) s **max{l(v), l(ui) + w(u,v)}**. Izračunaj **max{l(v)}**, te odredi ui+1 kao onaj vrh za koji se taj **maksimum**
3. Nema promjene postize. Stavi...

14.

(Odgovor: b0ysee) Ovaj zadatak rijesava se isto kao i 11. samo sto treba koristiti ovaj promijenjeni algoritam, tj. uvijek traziti maksimalnu vrijednost.

Trenutno mi se stvarno ne da ispisivati, samo cu napisati put:

A -> C -> B -> E -> D -> F -> G

15.

(Odgovor: b0ysee) Placeholder, valjda ce mi se dati rijesiti, ovo je samo drilanje postupaka koji su vec navedeni. Dobije se graf gradova i treba naci sve moguce kombinacije najkracih puteva izmedju dvaju tocaka.

17.

Za ovaj graf se lako da vidjeti da je skoro eulerovski jer ima tocno dva vrha neparnog stupnja. To nam olaksava problem kineskog postara ekstremno, zato sto onda mozemo naci eulerovsku stazu (tako da postar svaki brid prodje samo jednom dok raznosi postu), a nakon toga ga vratimo u sjediste najkracim mogucim putem (cak i u knjizici kazu da ako dobijemo takav graf da se on svodi na: nalazenje eulerovske staze + Dijsktrin algoritam kako bi se sto brze vratili u bazu).

Eulorovskih staza ovdje ima koliko hocete, samo da podsjetim - to je put kojem su svi bridovi razliciti. Evo, ja cu ici (po tezinama bridova) ovako: 7 -> 2 -> 6 -> 3 -> 10 -> 3 -> 4 -> 2 -> 8 (Sve skupa: 45).
Za nalazenje puta natrag kod ovako jedostavnog grafa cijeli Dijsktrin algoritam stvarno nije potreban, biti ce dovoljno samo se posluziti tim principima i naci ga odoka. Prvo se vratimo bridom tezine 8 natrag. Sada imamo 3 izbora: gornja staza iznosi (3+6+2) 11, srednja iznosi 10, a doljnja iznosi (2+4+3) 9, sto znaci da cemo izabrati donju. Tome pridodamo jos i ovih zadnjih 7 i imamo: 45 + 8 + 9 + 7 = 69, sto je i najkraci put kojim kineski postar moze ici.

18.

Samo da potvrdim odgovor na 18. Tocno je 34, pitao sam jucer Pavcevica i on je potvrdio da ako dobijemo, za kineski problem postaram, graf koji je ili lanac ili stablo, tada treba svaki brid uciniti dvostrukim da bi se problem mogao rijesiti, a onda mora proci put koji iznosi |E(G)| * 2 = 17*2.

19.

Pohlepni algoritam ovdje daje rjesenje.
Pohlepni algoritam je nacin trazenja najkraceg puta po grafu na nacin da gledamo koji nam je u tom trenutku put dalje najbolji, tj. u svakom vrhu ponovno gledamo kud cemo dalje i biramo najbolji izbor.

Krenimo iz vrha A, ici cemo u vrh D zato sto je tezina tog brida 3, nakon toga idemo u vrh E jer ovdje imamo tezinu 2, pa zatim u vrh C jer je tezina opet 2, pa zatim u vrh B jer je tezina 3 i nakon toga se vracamo u vrh A (preko brida tezine 4).
Ukratko: A -> D -> E -> C -> B -> A (tj. 3+2+2+3+4 = 14)

Sad, kako najbrze provjeriti jesmo li stvarno isli najboljim putem. Nakon sto sam dosta gledao taj problem cini mi se da je najjednostavnije pogledati sve tezine bridova i popisati ih u jednom redu. Nakon toga pogledamo koliko bridova moramo proci, te vidimo u kolikoj mjeri koristimo bridove najmanje tezine.

Konkretno za ovaj primjer,
imamo bridove tezina: 2,2,3,3,4,4,5,6,7,8; ako pogledamo nas put gore mi smo koristili bridove tezina 2,2,3,3 i 4 sto je upravo najmanje sto smo i mogli! :)

20.

Kako bi sto brze dosli do rjesenja mozemo opet krenuti s pohlepnim algoritmom (ali trazeci najvece vrijednosti), pa pogledati zadovoljava li to nase apetite. :)

Krenimo iz A:
A -> C (tezina 6)
C -> D (tezina 4)
D -> B (tezina 7)
B -> E (tezina 8)
E -> A (tezina 5)
Tezine bridova u nasem grafu: 2,2,3,3,4,4,5,6,7,8, a mi smo koristili 4,5,6,7 i 8 - problem rjesen! :)