Respostes per a la modalitat **A**

1. Es vol ordenar d nombres diferents compresos entre 1 i n. Per a fer-ho s'usa un vector de n booleans que s'inicialitzen primer a *false*. A continuació es recorren els d nombres canviant els valors de l'element del vector de booleans corresponent al seu nombre a *true*. Finalment es recorre el vector de booleans escrivint els índexs dels elements del vector de booleans que són *true*. És aquest algorisme més ràpid (asimptòticament) que el *mergesort*?

   [(a)] Només si $d \log d > k\,n$ (on k és una constant que depén de la implementació)

   (b) Sí, ja que el *mergesort* és $O(n \log n)$ i est és $O(n)$

   (c) No, ja que aquest algorisme ha de recórrer diverses vegades el vector de booleans.

2. En una quadrícula es vol dibuixar el contorn d'un quadrat de n caselles de costat. Quina serà la complexitat temporal del millor algorisme que pugui existir?

   (a) $O(n^2)$

   [(b)] $O(n)$

   (c) $O(\sqrt{n})$

3. La complexitat en el millor dels casos d'un algorisme de *ramificació i poda* ...

   (a) ... és sempre exponencial respecte del nombre de decisions a prendre.

   (b) ... sol ser polinòmica respecte del nombre d'alternatives per cada decisió.

   [(c)] ... pot ser polinòmica respecte del nombre de decisions a prendre.

4. La solució recursiva ingènua (però correcta) a un problema d'optimització crida més d'una vegada a la funció amb els mateixos paràmetres. Una de les següents tres afirmacions és falsa.

   [(a)] Es pot millorar l'eficiència de l'algorisme convertint l'algorisme recursiu directament en iteratiu sense canviar el seu funcionament bàsic.

   (b) Es pot millorar l'eficiència de l'algorisme guardant en una taula el valor retornat per a cada conjunt de paràmetres de cada trucada quan aquesta es produeix per primera vegada.

   (c) Es pot millorar l'eficiència de l'algorisme definint per endavant l'ordre en el qual s'han de calcular les solucions als subproblemes i omplint una taula en aquest ordre.

5. Quan s'usa un algorisme voraç per abordar la resolució d'un problema d'optimització per selecció discreta (és a dir, un problema per al qual la solució consisteix a trobar un subconjunt del conjunt d'elements que optimitza una determinada funció), quina d'aquestes tres coses és impossible que ocórrega?

   [(a)] Que es reconsidere la decisió ja presa anteriorment respecte a la selecció d'un element a la vista de la decisió que s'ha de prendre en l'instant actual.

   (b) Que l'algorisme no trobe cap solució.

   (c) Que la solució no siga l'òptima.

6. Quin dels següents algorismes proveiria una fita pessimista per al problema de trobar el camí mes curt entre dues ciutats (se suposa que el graf és connex)?

   [(a)] Calcular la distància recorreguda movent-se a l'atzar pel graf fins a arribar (per atzar) a la ciutat de destinació.

   (b) Calcular la distància geomètrica (en línia recta) entre la ciutat d'origen i la de destinació.

   (c) Per a totes les ciutats a les quals es pot arribar en un pas des de la ciutat inicial, sumar la distància a aquesta ciutat i la distància geomètrica fins a la ciutat de destinació.

7. Per a quin d'aquests problemes d'optimització existeix una solució voraç?

   [(a)] L'arbre de recobriment mínim per a un graf no dirigit amb pesos.

   (b) El problema de la motxilla discreta.

   (c) El problema de l'assignació de cost mínim de n tasques a n treballadors quan el cost d'assignar la tasca i al treballador j, c_{ij} està tabulat en una matriu.

8. Un problema de grandària n pot transformar-se en temps $O(n^2)$ en un altre de grandària $n-1$. D'altra banda, la solució al problema quan la talla és 1 requereix un temps constant. Quina d'aquestes classes de cost temporal asimptòtic és la més ajustada?

   (a) $O(2^n)$

   (b) $O(n^2)$

   [(c)] $O(n^3)$

9. Quin d'aquests problemes té una solució eficient que usa *programació dinàmica*?

   (a) El problema de l'assignació de tasques.

   (b) La motxilla discreta sense restriccions addicionals.

   [(c)] El problema del canvi.

10. Si un problema d'optimització ho és per a una funció que pren valors continus ...

    [(a)] La programació dinàmica recursiva pot resultar molt més eficient que la programació dinàmica iterativa quant a l'ús de memòria.

    (b) La programació dinàmica iterativa sempre és molt més eficient que la programació dinàmica iterativa quant a l'ús de memòria.

    (c) L'ús de memòria de la programació dinàmica iterativa i de la programació dinàmica recursiva és el mateix independentment de si el domini és discret o continu.

11. En resoldre el problema del viatjant de comerç mitjançant tornada arrere, quina d'aquestes fites optimistes s'espera que pode millor l'arbre de recerca?

    [(a)] S'ordenen les arestes restants de menor a major distància i es calcula la suma de les k arestes més curtes, on k és el nombre de salts que ens queden per donar.

    (b) Es multiplica k per la distància de l'aresta més curta que ens queda per considerar, on k és el nombre de salts que ens queden per donar.

    (c) Es resol la resta del problema usant un algorisme voraç que afegeix cada vegada al camí el vèrtex més proper a l'últim afegit.

12. La versió de *Quicksort* que utilitza com a pivot l'element del vector que ocupa la primera posició ...

    (a) ...no presenta cas millor i pitjor per a instàncies de la mateixa grandària.

    [(b)] ... es comporta pitjor quan el vector ja està ordenat.

    (c) ... es comporta millor quan el vector ja està ordenat.

13. El programa següent resol el problema de tallar un tub de longitud n en segments de longitud entera entre 1 i n de manera que es maximitze el preu d'acord amb una taula que dóna el preu per a cada longitud, però en falta un tros. Què hauria d'anar en lloc de **XXXXXXX**?

```
void fill(price r[]) {
for (index i=0;i<=n;i++) r[i]=-1;
}

price cutrod(price p[], r[], length n) {
price q;
if (r[n]>=0) return r[n];
if (n==0) q=0;
else {
       q=-1;
       for (index i=1;i<=n;i++)
       q=max(q,p[i]+cutrod(XXXXXXX));
}
r[n]=q;
return q;
}
```

(a) `p,r,n-i`
(b) `p,r-1,n`
(c) `p,r,n-r[n]`

14. Si $f(n) \in O(n^3)$, pot passar que $f(n) \in O(n^2)$?
    (a) És perfectament possible, ja que $O(n^2) \subset O(n^3)$
    (b) Només per a valors baixos de n
    (c) No, perquè $n^3 \notin O(n^2)$

15. Digueu quin d'aquests tres algorismes no és un algorisme de "divideix i venceràs"
    (a) L'algorisme de Prim
    (b) Quicksort
    (c) Mergesort

16. La complexitat temporal en el millor dels casos...
    (a) ... és una funció de la grandària o talla del problema que ha d'estar definida per tots els possibles valors d'aquesta.
    (b) ... és el temps que tarda l'algorisme a resoldre el problema de grandària o talla més petita que se li pot presentar.
    (c) Les dues anteriors són certes.

17. L'ús de funcions de fita en ramificació i poda...

    (a) ... transforma en polinòmiques complexitats que abans eren exponencials.

    [(b)] ... pot reduir el nombre d'instàncies del problema que pertanyen al cas pitjor.

    (c) ... garanteix que l'algorisme serà més eficient davant qualsevol instància del problema.

18. Donades les funcions següents:

```
// Precondició: { 0 <= i < v.size(); i < j <= v.size() }
unsigned f( const vector<unsigned>&v, unsigned i, unsigned j ) {
  if( i == j+1 )
    return v[i];
  unsigned sum = 0;
  for( unsigned k = 0; k < j - i; k++ )
    sum += f( v, i, i+k+1) + f( v, i+k+1, j );
  return sum;
}

unsigned g( const vector<unsigned>&v ) {
  return f( v, v.begin(), v.end() );
}
```

    Es vol reduir la complexitat temporal de la funció `g` usant programació dinàmica iterativa. Quina en seria la complexitat espacial?

    [(a)] quadràtica

    (b) cúbica

    (c) exponencial

19. Quin d'aquests tres problemes d'optimització no té, o no se li'n coneix, una solució voraç (*greedy*) que siga òptima?

    (a) El problema de la motxilla contínua o amb fraccionament.

    [(b)] El problema de la motxilla discreta.

    (c) L'arbre de cobertura de cost mínim d'un graf connex.

20. En els algorismes de *ramificació i poda* ...

    [(a)] Una fita optimista és necessàriament un valor insuperable; si no fóra així, es podria podar el node que condueix a la solució òptima.

    (b) Una fita optimista és necessàriament un valor abastable; si no és així, no està garantit que es trobe la solució òptima.

    (c) Una fita pessimista és el valor que com a màxim aconsegueix qualsevol node factible que no és l'òptim.

21. Una d'aquestes tres situacions no és possible:

    [(a)] $f(n) \in \Omega(n^2)$ i $f(n) \in O(n)$

    (b) $f(n) \in O(n)$ i $f(n) \in \Omega(1)$

    (c) $f(n) \in O(n)$ i $f(n) \in O(n^2)$

22. En els algorismes de *ramificació i poda*, el valor d'una fita pessimista és més gran que el valor d'una fita optimista? (s'entén que ambdues fites s'apliquen sobre el mateix node)

    (a) No, mai no és així.

    (b) En general sí, si es tracta d'un problema de maximització, encara que en ocasions tots dos valors poden coincidir.

    [(c)] En general sí, si es tracta d'un problema de minimització, encara que en ocasions tots dos valors poden coincidir.

23. Un d'aquests tres problemes no té una solució eficient que seguisca l'esquema de programació dinàmica

    [(a)] El problema de les torres d'Hanoi

    (b) El problema de la motxilla discreta.

    (c) El problema de tallar un tub de longitud n en segments de longitud sencera entre 1 i n de manera que es maximitze el preu d'acord amb una taula que dóna el preu per a cada longitud.

24. Donat un problema d'optimització, el mètode voraç...

    (a) ... sempre obté la solució òptima.

    (b) ... sempre obté una solució factible.

    [(c)] ... garanteix la solució òptima només para determinats problemes.

25. Donat un problema d'optimització qualsevol, l'estratègia de *tornada arrere* garanteix la solució òptima?

    (a) Sí, ja que aquest mètode analitza totes les possibilitats.

    [(b)] És condició necessària que el domini de les decisions siga discret o discretizable i que el nombre de decisions a prendre estiga fitat.

    (c) Sí, sempre que el domini de les decisions siga discret o discretizable i a més s'usen mecanismes de poda basats en la millor solució fins al moment.

26. Garanteix l'ús d'una estratègia "divideix i venceràs" l'existència d'una solució de complexitat temporal polinòmica a qualsevol problema?

    [(a)] No

    (b) Sí, en qualsevol cas.

    (c) Sí, però sempre que la complexitat temporal conjunta de les operacions de descomposició del problema i la combinació de les solucions siga polinòmica.

27. Si per resoldre un mateix problema usem un algorisme de tornada arrere i el modifiquem mínimament per convertir-ho en un algorisme de ramificació i poda, què canviem realment?

    [(a)] L'algorisme pot aprofitar millor les fites optimistes.

    (b) Canviem la funció que donem a la fita pessimista.

    (c) La comprovació de les solucions factibles: en ramificació i poda no és necessari ja que només genera nodes factibles.

28. Es vol trobar el camí mes curt entre dues ciutats. Per a açò es disposa d'una taula amb la distància entre els parells de ciutats entre les quals hi ha carretera o un valor sentinella (per exemple, -1) si no n'hi ha; per això, per anar de la ciutat inicial a la final és possible que calga passar per diverses ciutats. També es coneixen les coordenades geogràfiques de cada ciutat i per tant la distància geomètrica (en línia recta) entre cada parell de ciutats. Es pretén accelerar la recerca d'un algorisme de *ramificació i poda* prioritzant els nodes vius (ciutats) que estiguen a menor distància geogràfica de la ciutat objectiu.

    (a) El nou algorisme sempre sera més ràpid.

    [(b)] El nou algorisme no garanteix que vaja a ser més ràpid per a totes les instàncies del problema possibles.

    (c) Aquesta estratègia no assegura que s'obtinga el camí mes curt.

29. La millora que en general aporta la programació dinàmica enfront de la solució ingènua s'aconsegueix gràcies al fet que ...

    [(a)] ... en la solució ingènua es resol moltes vegades un nombre relativament reduït de subproblemes diferents.

    (b) ... en la solució ingènua es resol poques vegades un nombre relativament gran de subproblemes diferents.

    (c) El nombre de vegades que es resolen els subproblemes no té res a veure amb l'eficiència dels problemes resolts mitjançant programació dinàmica.

30. La millor solució que es coneix per al problema de la motxilla contínua segueix l'esquema ...

    (a) ... *voraç.*

    [(b)] ... *divideix i venceràs.*

    (c) ... *ramificació i poda.*

31. Un algorisme recursiu basat en l'esquema *divideix i venceràs* ...

    [(a)] ... serà més eficient com més equitativa siga la divisió en subproblemes.

    (b) ... no tindrà mai una complexitat exponencial.

    (c) Les dues anteriors són certes.

32. En l'esquema de tornada arrere, els mecanismes de poda basats en la millor solució fins al moment...

    [(a)] ... poden eliminar solucions parcials que són factibles.

    (b) ... garanteixen que no s'explorarà mai tot l'espai de solucions possibles.

    (c) Les dues anteriors són veritables.

33. Digueu quina d'aquestes tres és la fita pessimista més ajustada al valor òptim de la motxilla discreta:

    (a) El valor de la motxilla contínua corresponent

    [(b)] El valor de la motxilla discreta que s'obté usant un algorisme voraç basat en el valor específic dels objectes

    (c) El valor d'una motxilla que conté tots els objectes encara que es passe del pes màxim permés

34. Siga la relació següent de recurrència

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 2T(\frac{n}{2}) + g(n) & \text{en un altre cas} \end{cases}$$

    Si $T(n) \in O(n^2)$, en quin d'aquests tres casos ens podem trobar?

    (a) $g(n) = n$

    (b) $g(n) = 1$

    [(c)] $g(n) = n^2$

35. Quan la descomposició recursiva d'un problema dóna lloc a subproblemes de grandària similar, quin esquema promet ser més apropiat?

    (a) Divideix i venceràs, sempre que es garantisca que els subproblemas no són de la mateixa grandària.

    [(b)] Programació dinàmica.

    (c) El mètode voraç

36. En l'esquema de *tornada arrere* l'ordre en el qual es van assignant els diferents valors a les components del vector que contindrà la solució...

    (a) ... és irrellevant si no s'utilitzen mecanismes de poda basats en la millor solució fins al moment.

    (b) ... pot ser rellevant si s'utilitzen mecanismes de poda basats en estimacions optimistes.

    [(c)] Les dues anteriors són certes.

37. Quan es resol el problema de la motxilla discreta usant l'estratègia de tornada arrere, pot ocórrer que es tarde menys a trobar la solució òptima si es prova primer a ficar-hi cada objecte abans de no ficar-l'hi?

    [(a)] Sí, però només si s'usen fites optimistes per podar l'arbre de recerca.

    (b) Sí, tant si s'usen fites optimistes per podar l'arbre de recerca com si no.

    (c) No, ja que en qualsevol cas s'han d'explorar totes les solucions factibles.

38. En un problema d'optimització, si el domini de les decisions és un conjunt infinit,

    (a) és probable que a través de programació dinàmica s'obtinga un algorisme eficaç que el resolga.

    (b) podrem aplicar l'esquema de tornada arrere sempre que es tracte d'un conjunt infinit numerable.

    [(c)] una estratègia voraç pot ser l'única alternativa.

39. El valor que s'obté amb el mètode voraç per al problema de la motxilla discreta és ...

    (a) ... una fita inferior per al valor òptim, però que mai coincideix amb aquest.

    [(b)] ... una fita inferior per al valor òptim que de vegades pot ser igual a aquest.

    (c) ... una fita superior per al valor òptim.

40. Siga A una matriu quadrada $n \times n$. Es tracta de buscar una permutació de les columnes tal que la suma dels elements de la diagonal de la matriu resultant siga mínima. Indiqueu quina de les següents afirmacions és falsa.

    [(a)] La complexitat temporal de la millor solució possible al problema és $O(n^2)$.

    (b) La complexitat temporal de la millor solució possible al problema és $O(n!)$.

    (c) Si es construeix una solució al problema basada en l'esquema de ramificació i poda, una bona elecció de fites optimistes i pessimistes podria evitar l'exploració de totes les permutacions possibles.