

Algoritmi și programare - Tema 5

Termen de realizare: săptămâna 28 ian - 1 feb 2013

Observații

- Toate problemele sunt obligatorii.
- Vor primi punctaj maxim soluțiile optime din punct de vedere al complexității.

Obiective

- Implementarea unor algoritmi de sortare și căutare
- Proiectarea rezolvării problemelor de enumerare
- Familiarizarea cu tehnica de programare backtracking

Probleme propuse

1. Scrabble

Fanii jocului Scrabble sunt obișnuiți cu *anagramele* (grupuri de cuvinte cu aceleași litere, dar în altă ordine), de exemplu: *acar*, *arac*, *cara*. Există, totuși, cuvinte care nu au acest atribut, adică indiferent cum sunt aranjate literele componente, nu se poate forma un alt cuvânt (exemplu: *mos*).

Asemenea cuvinte sunt numite *ananagrame*. Bineînțeles că aceste definiții depind de domeniul în care lucrăm. Un asemenea domeniu poate fi întreg dicționarul limbii române. Putem restrânge domeniul. Spre exemplu, domeniul muzical, în care *nota* devine o *ananagramă relativă* la acest domeniu, pentru că *tona* nu face parte din acest domeniu.

Scrieți un program care citește un dicționar dintr-un anumit domeniu memorat în fișierul de intrare `scrabble.in`, și care determină toate ananagramele relative. Cuvintele formate dintr-o singură literă sunt ananagrame, deoarece ele nu pot fi "rearanjate". Ananagramele găsite vor fi scrise în fișierul de ieșire `scrabble.out`, fiecare pe câte o linie, în ordine lexicografică.

Precizări

- Un cuvânt are cel mult 20 de litere ale alfabetului englez.
- Un dicționar nu conține mai mult de 1000 de cuvinte.
- Liniile fișierului nu au mai mult de 80 de caractere.
- Orice cuvânt se găsește în întregime pe o linie.
- Cuvintele sunt separate prin spații.
- `tieD` și `EdiT` sunt anagrame.

Exemplu

<code>scrabble.in</code>	<code>scrabble.out</code>
<code>ladder came tape soon leader acme RIDE lone Dreis peat ScAlE orb eye Rides dealer NotE derail LaCeS drIed noel dire Disk mace Rob dries</code>	<code>Disk NotE derail drIed eye ladder soon</code>

2. Partiții

Se consideră mulțimea $A(n) = \{1, 2, \dots, n-1, n\}$, unde n este un număr natural nenul.

Cerință

Scrieți o funcție care generează toate partițiile mulțimii $A(n)$.

Executați programul pentru valori din ce în ce mai mari ale lui n . Ce observați?

Precizări

- Numărul n va fi dat ca parametru funcției `main`.
- În caz de succes, rezultatul va fi scris în `stdout`.
- În caz de eroare, un mesaj va fi scris în `stderr`.

Exemple

<code>argv[1]</code>	<code>stdout</code>	<code>stderr</code>
2	<code>{{1}, {2}}</code> <code>{{1, 2}}</code>	
3	<code>{{1}, {2}, {3}}</code> <code>{{1}, {2, 3}}</code> <code>{{1, 2}, {3}}</code> <code>{{1, 3}, {2}}</code> <code>{{1, 2, 3}}</code>	
h		<code>impossible</code>
		<code>impossible</code>

3. Bancnote

Se consideră un număr natural nenul s care reprezintă o sumă de bani și n tipuri de bancnote, având valorile naturale nenule $v_0, v_1, \dots, v_{n-2}, v_{n-1}$. Dispunem de m_0 bancnote de tipul 0, m_1 bancnote de tipul 1, ..., m_{n-2} bancnote de tipul $n-2$, m_{n-1} bancnote de tipul $n-1$.

Cerință

Găsiți toate modalitățile de plată a sumei s folosind bancnotele.

Nu se face distincție între două modalități de plată folosind bancnote de același tip.

Exemplu

<code>intrare</code>	<code>ieșire</code>
<code>s = 4</code> <code>n = 2</code> <code>v₀ = 1, m₀ = 4</code> <code>v₁ = 2, m₁ = 2</code>	<code>1+1+1+1</code> <code>1+1+2</code> <code>2+2</code>

4. Prime

Se consideră un număr natural nenul n .

Cerință

Scrieți o funcție care generează toate posibilitățile de scriere a lui n ca sumă de numere prime.

Exemplu

intrare	ieșire
8	2+2+2+2 2+3+3 3+5

5. Cuvinte

Se consideră n cuvinte de aceeași lungime și m cuvinte de lungime k .

Cerință

Scrieți un program care generează toate modalitățile de alegere a k cuvinte din cele n , astfel încât, scriind cuvintele unul sub altul, să se formeze pe prima și ultima coloană cuvinte din cele m .

Precizări

- Ordinea de citire din fișierul de intrare este: n , m , k , n cuvinte, m cuvinte.
- Soluțiile vor fi scrise într-un fișier de ieșire.

Exemplu

intrare	ieșire (două soluții posibile)
19 4 5 pac toc orb cum pom imi par cat act ele tic itu cot dia lin tac meu rau aur micut pitic prada curat	pom imi toc itu cot pac rau aur dia act