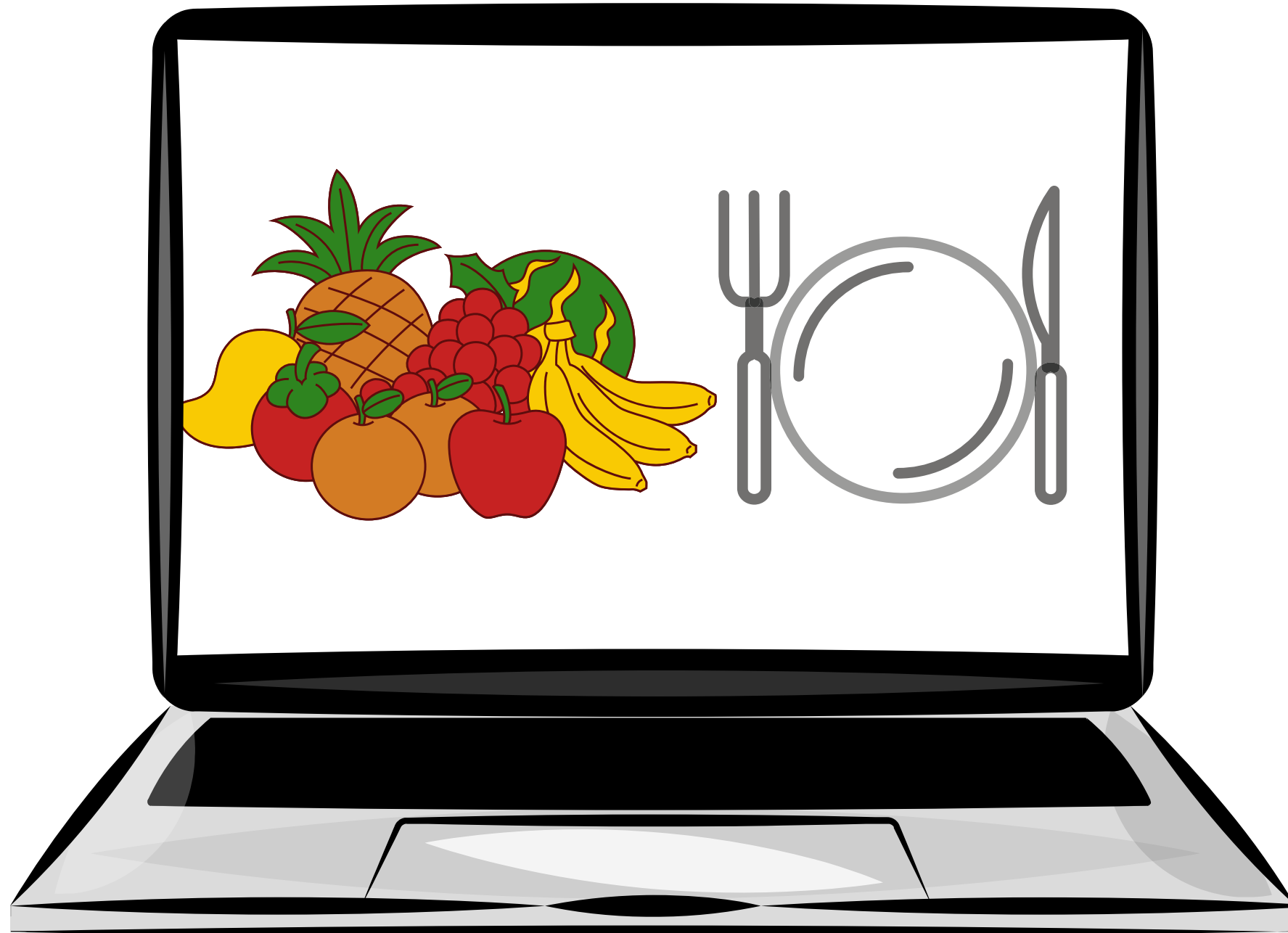


# TRANSDISCIPLINARITATEA METODEI BACKTRACKING



Coordonatori:

Măcelaru Mara

Barkoczi Nadia

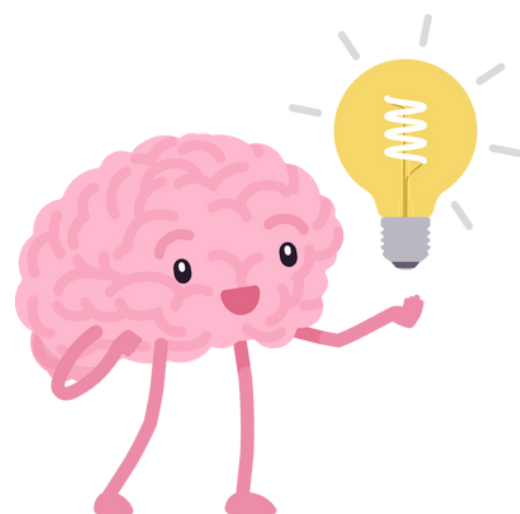
Proiect relaizat de:

Dragoș Anamaria Monica și Pop Antonio

# METODA BACKTRACKING

Unele caracteristici ale materialelor didactice prezentate sunt:

- Pot fi adaptate pentru a fi utilizate cu sau fără ajutorul profesorului;
- Este posibil să fie utilizate individual sau în grup;
- Sunt versatile, pot fi proiectate pentru diferite contexte;
- Permit elevului să fie capabil să dezvolte strategii pentru a evalua, planifica și organiza propria învățare.





# SCOPUL LECȚIEI

La finalul lecției, elevii vor fi capabili:

- Să explice conceptele de aranjamente, permutări și combinări
- Să utilizeze eficient metoda de backtracking pentru a implementa conceptele
- Să sublinieze legătura dintre matematică și informatică prin aplicarea interdisciplinarității în probleme practice
- Să construiască algoritmi pentru rezolvarea problemelor practice

# UTILIZAREA BACKTRACKINGULUI ÎN VIAȚA DE ZI CU ZI

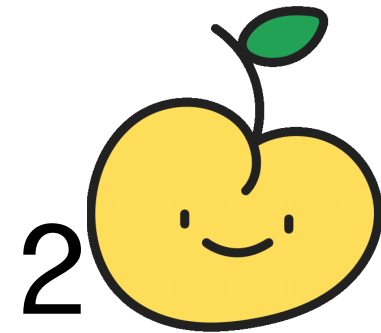
1. Utilizând metoda backtracking putem genera toate modalitățile de a prepara o băutură din câte 3 tipuri de fructe distincte din mulțimea: {căpșune, mere, pere, pepene roșu, portocale}.



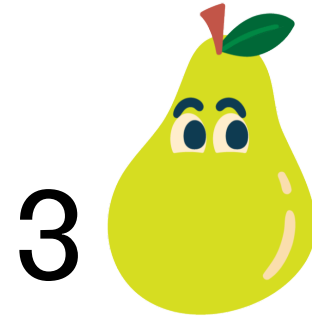
CĂPȘUNE



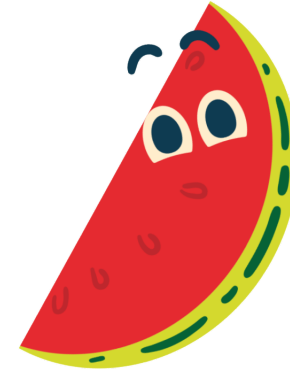
MERE



PERE



PEPENE ROȘU

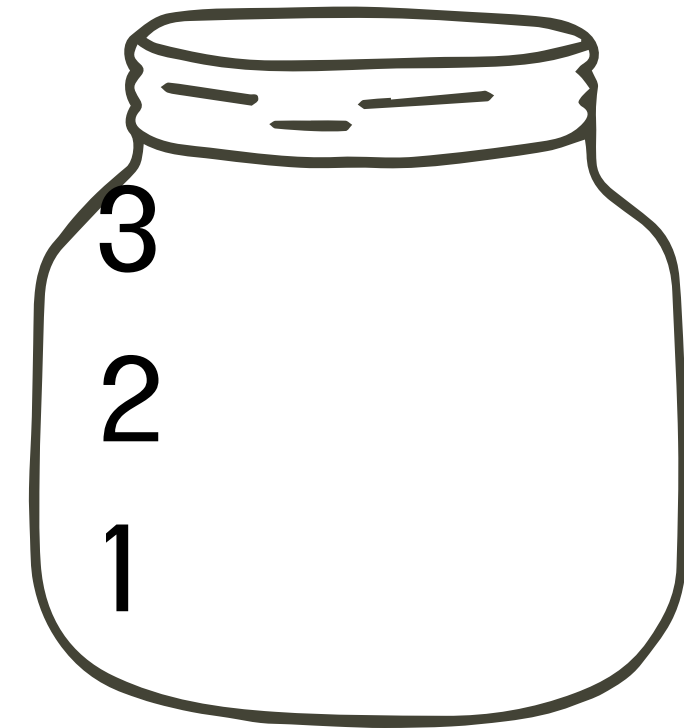
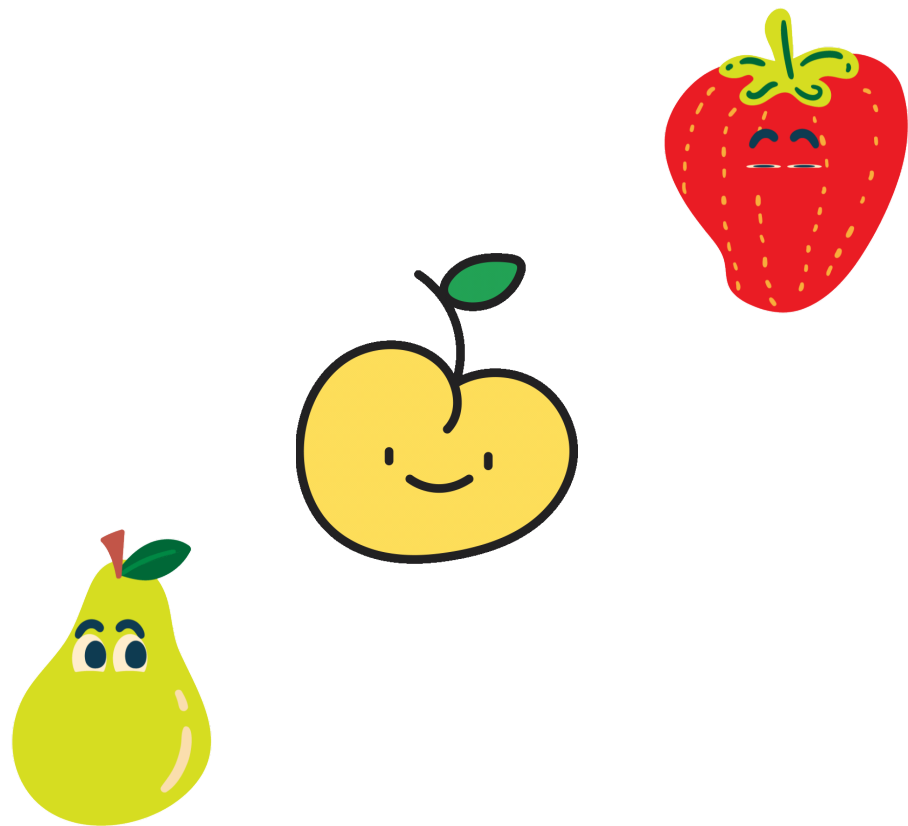


PORTOCALE

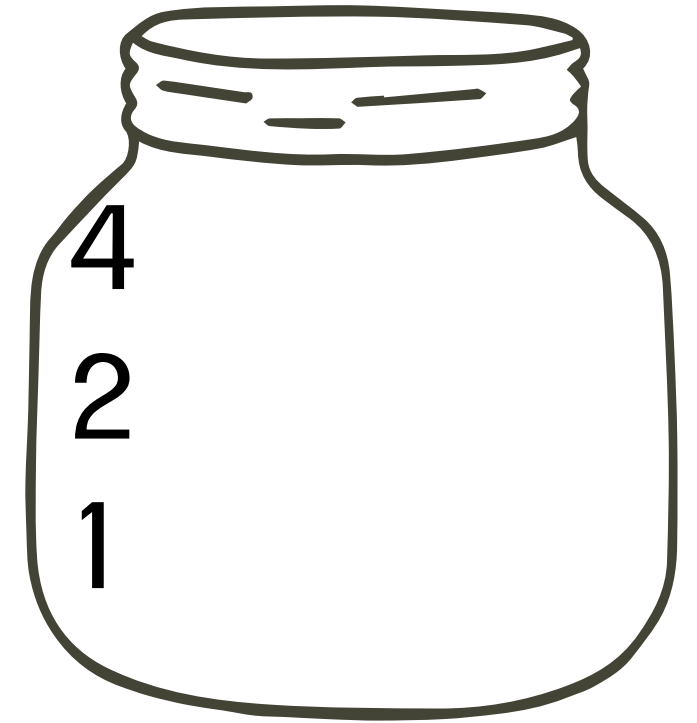
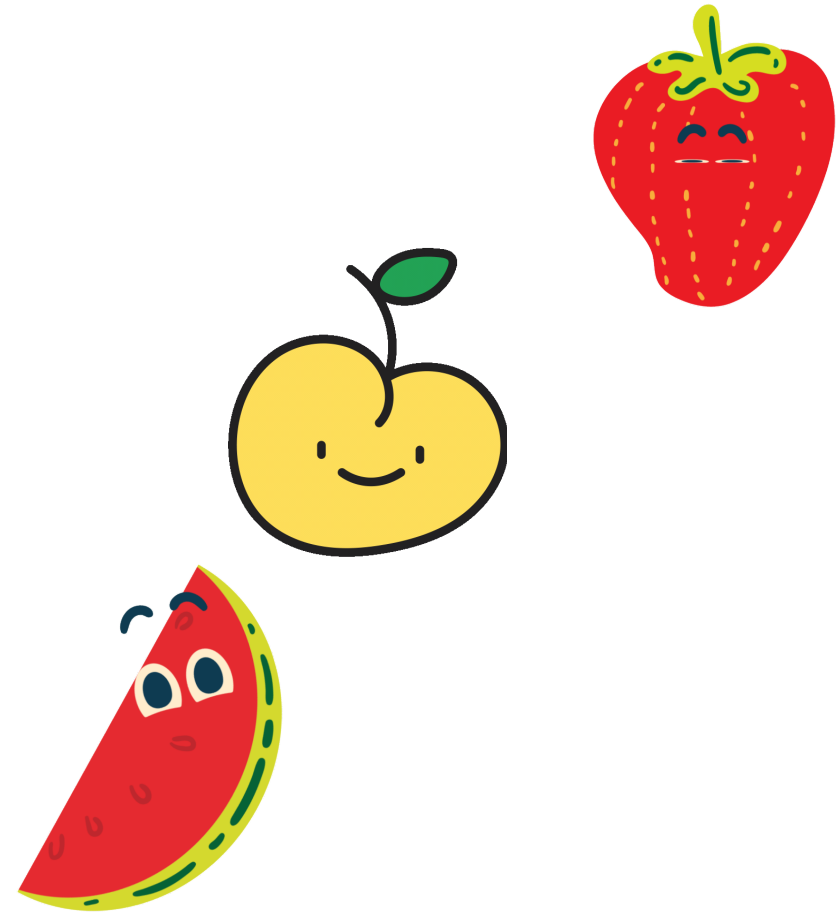


Primele soluții DE CÂTE 3 FRUCTE generate sunt, în această ordine:

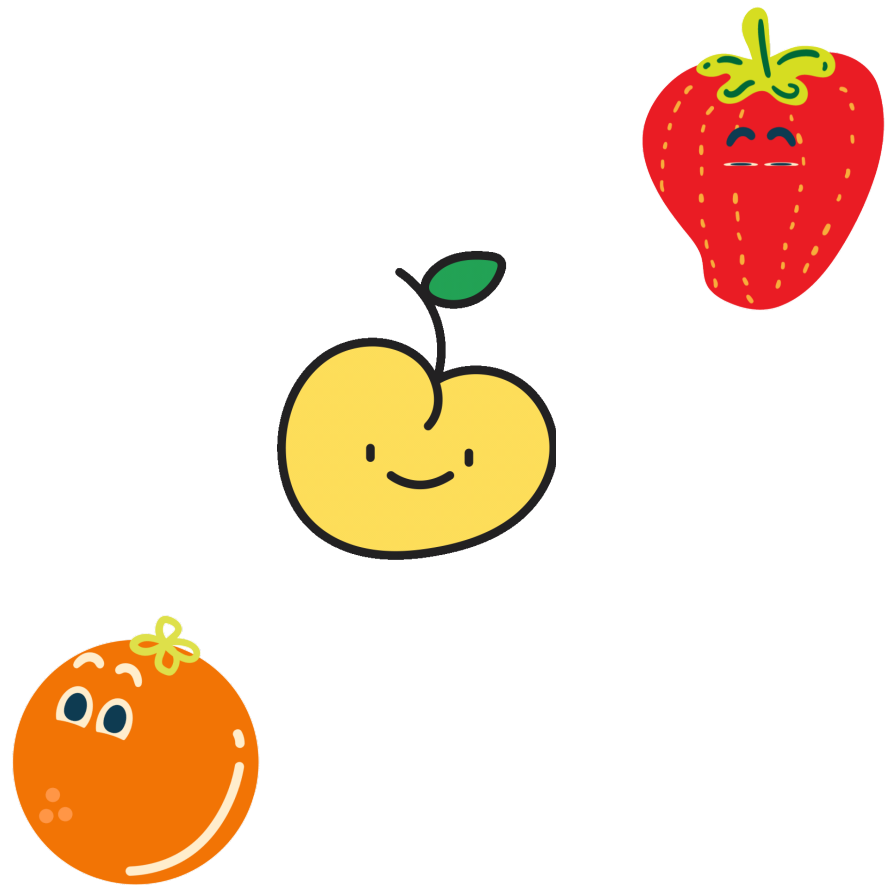
Prima soluție:



A doua soluție

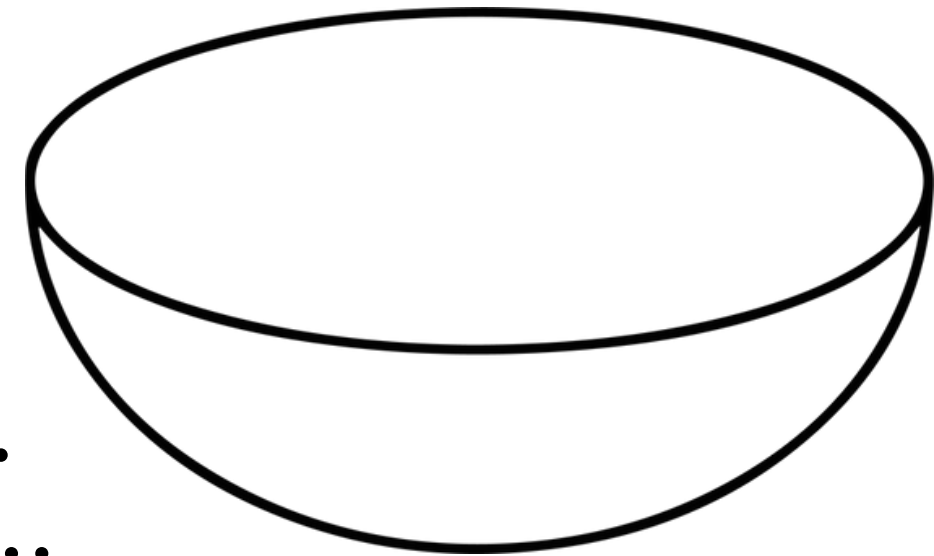
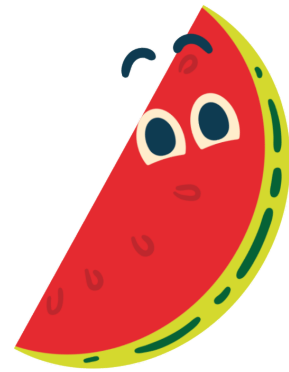
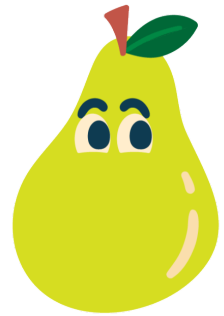
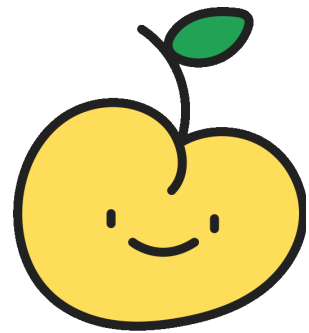


A treia soluție



Interdisciplinaritatea se remarcă prin faptul că, pentru a calcula numărul de soluții, folosim combinațiile.

Pentru a calcula soluțiile, vom calcula **COMBINĂRI DE 5 LUATE CÂTE 3**.



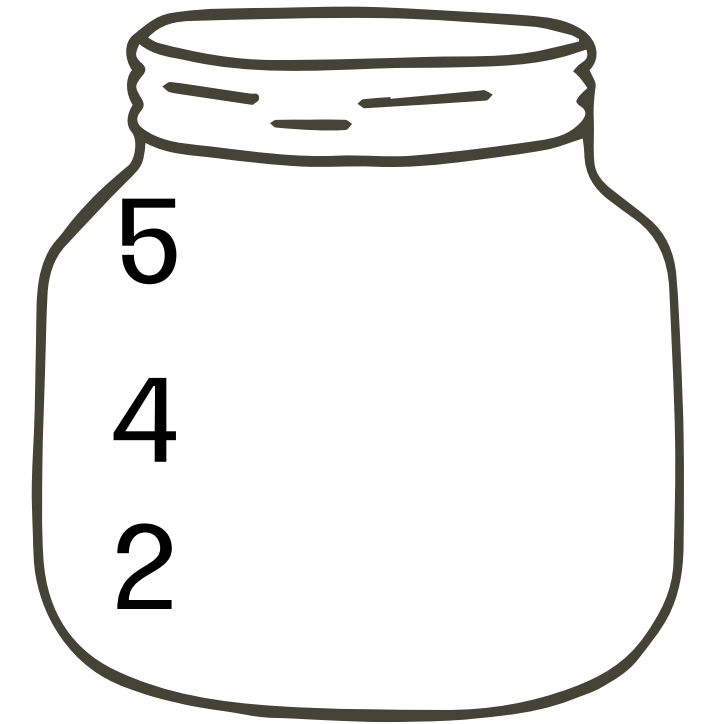
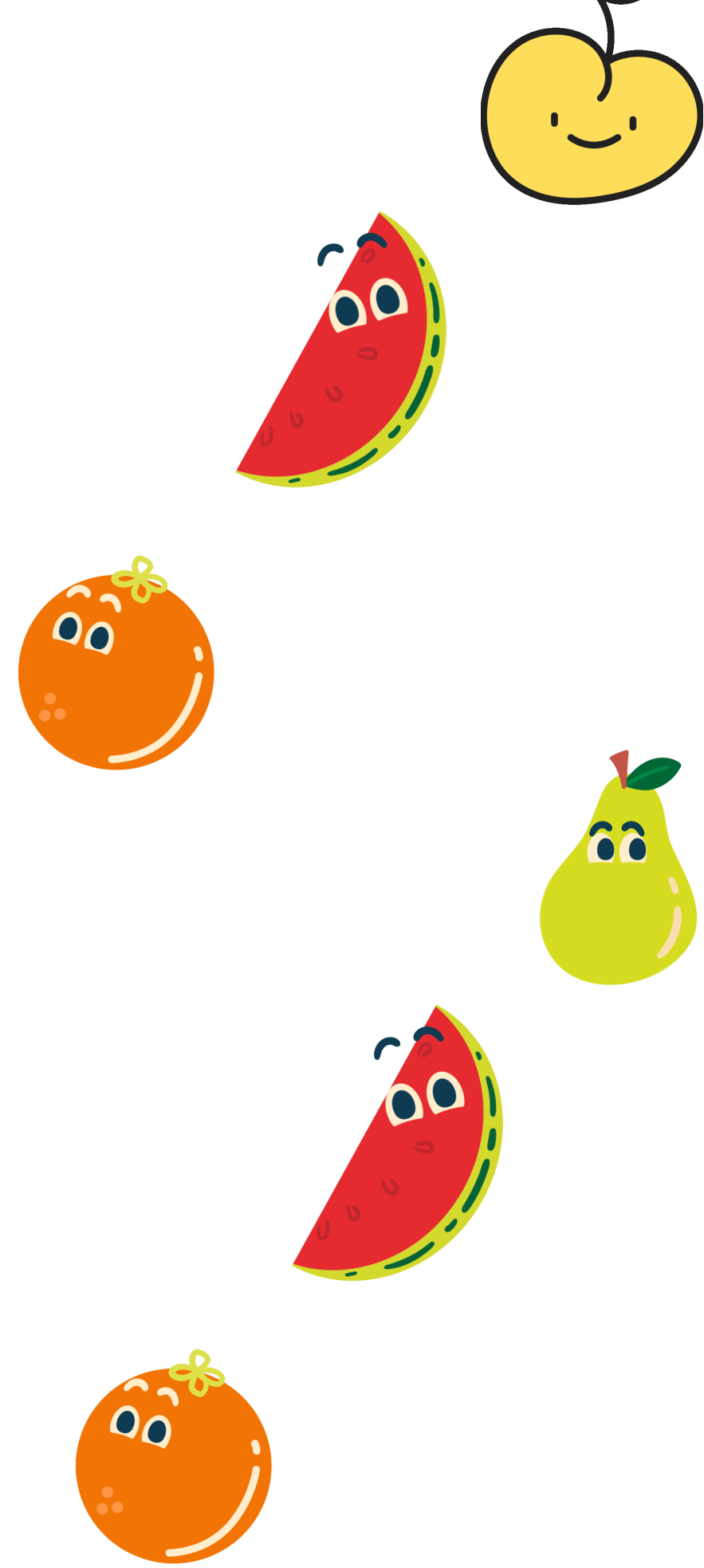
Am dat un exemplu de grupare câte 3 a elementelor.

După efectuarea calculelor, se obțin în total 10 soluții.

Vom ilustra în continuare ultimele două soluții.



Ultimele două soluții obținute sunt :

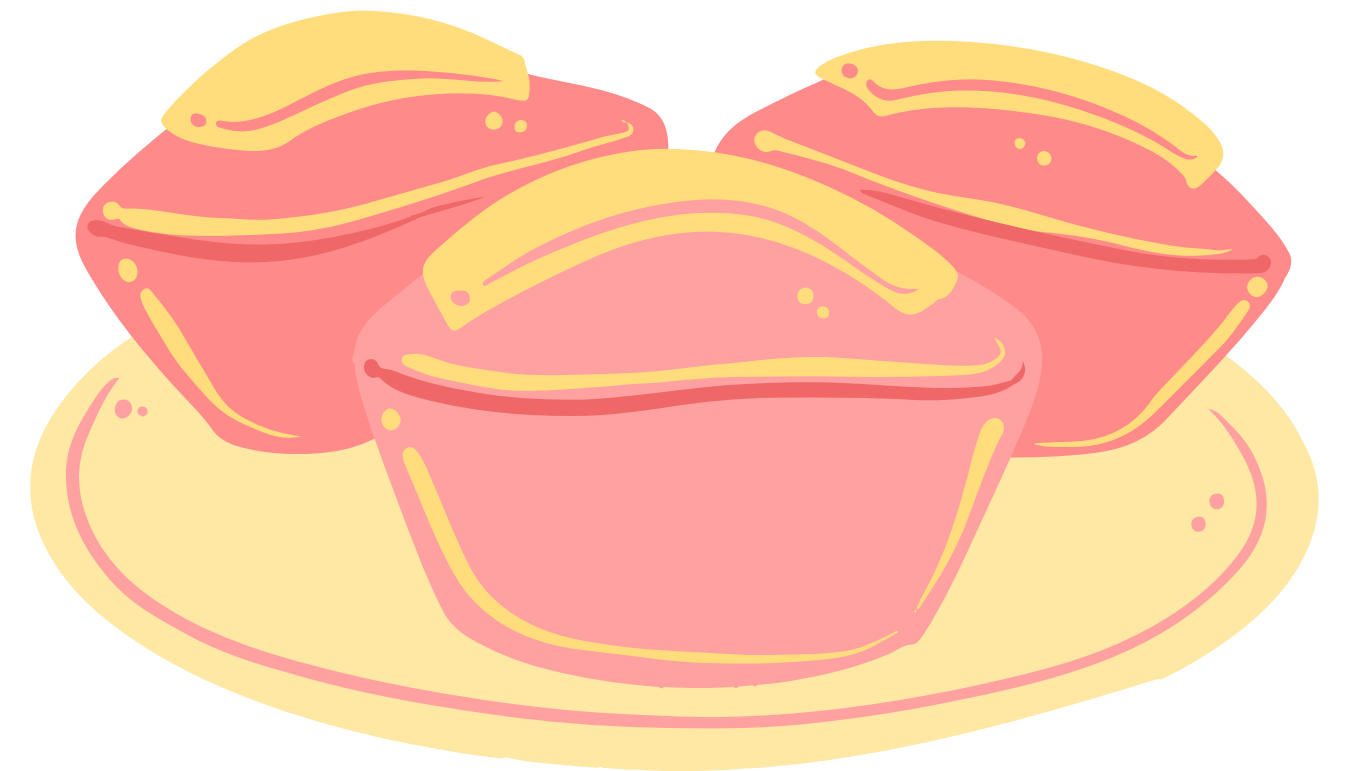




Spre deosebire de prima problemă, putem întâlni și probleme de backtracking în care să avem o condiție pe baza căreia se generează soluțiile.

2. Utilizând metoda backtracking, se generează toate meniurile care cuprind câte patru feluri distincte de mâncare din mulțimea {aperitiv, desert, legume, paste, salată, supă}.

Avem condiția ca primul fel servit să fie aperitivul, iar ultimul desertul.



Aperitiv

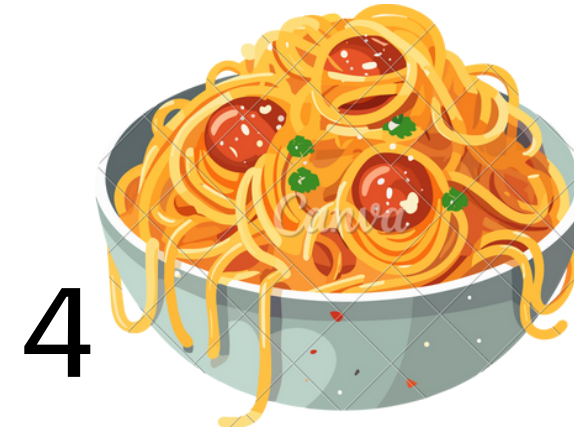
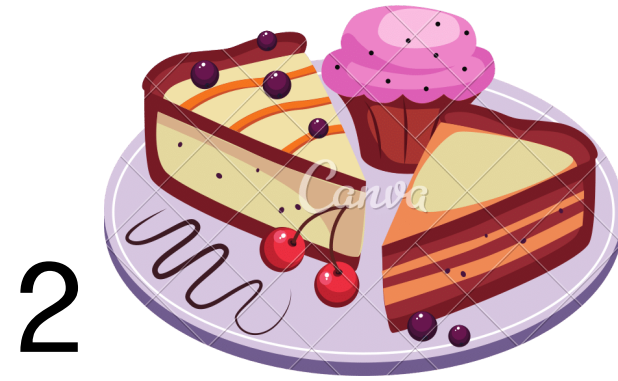
Desert

Legume

Paste

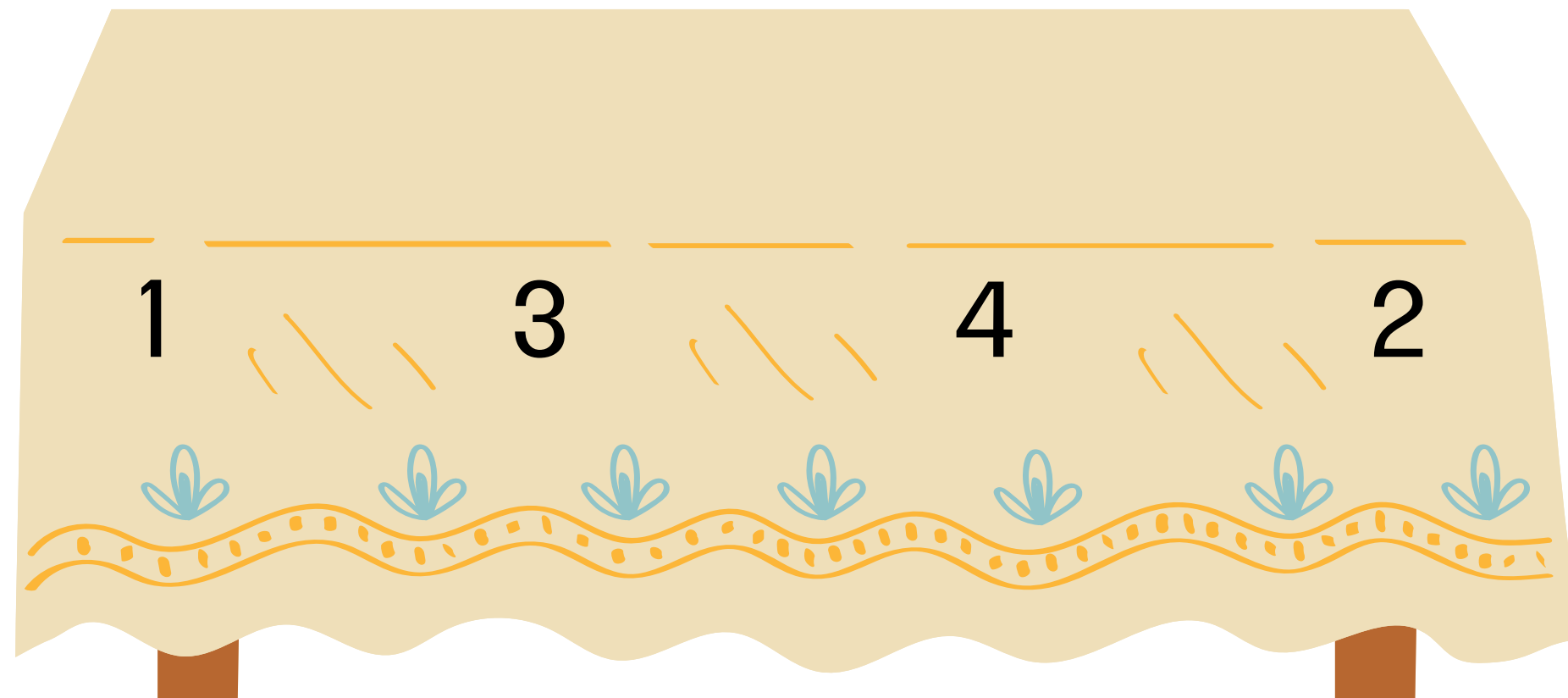
Salată

Supă

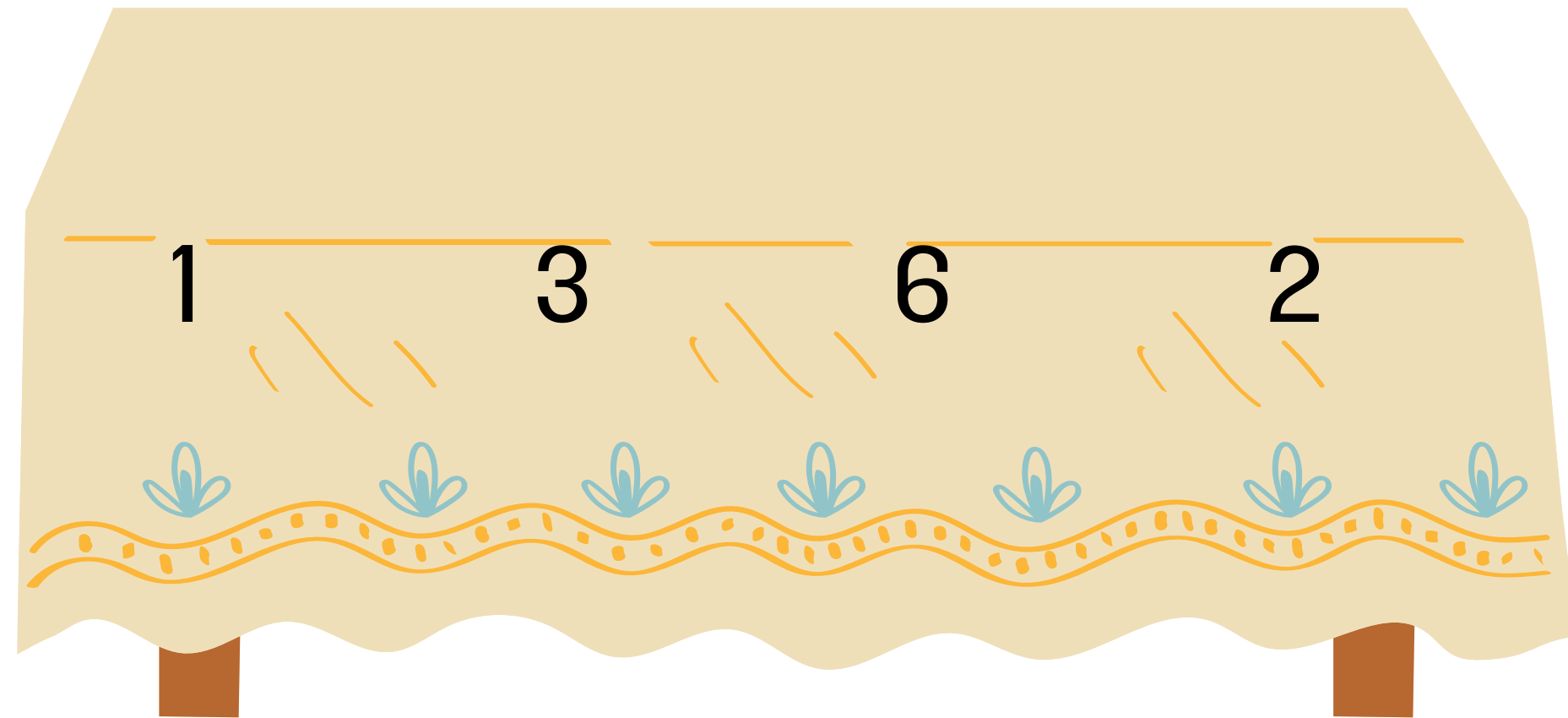
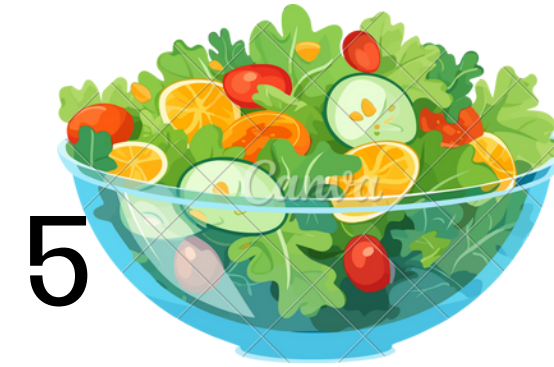
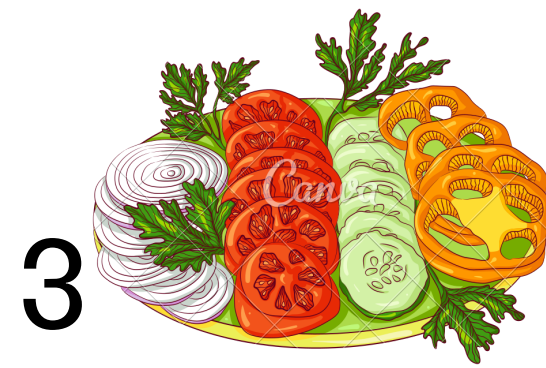
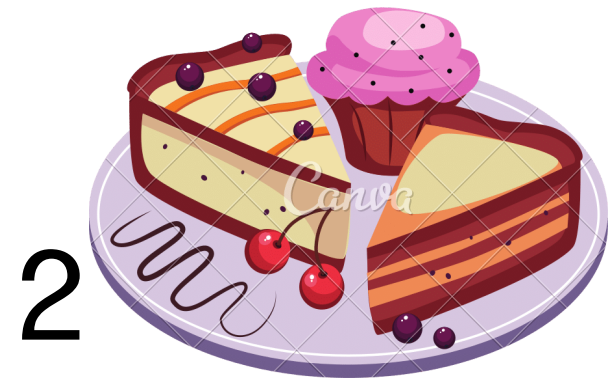


Primele soluții DE CÂTE 4 FELURI generate sunt, în această ordine:

Prima soluție:



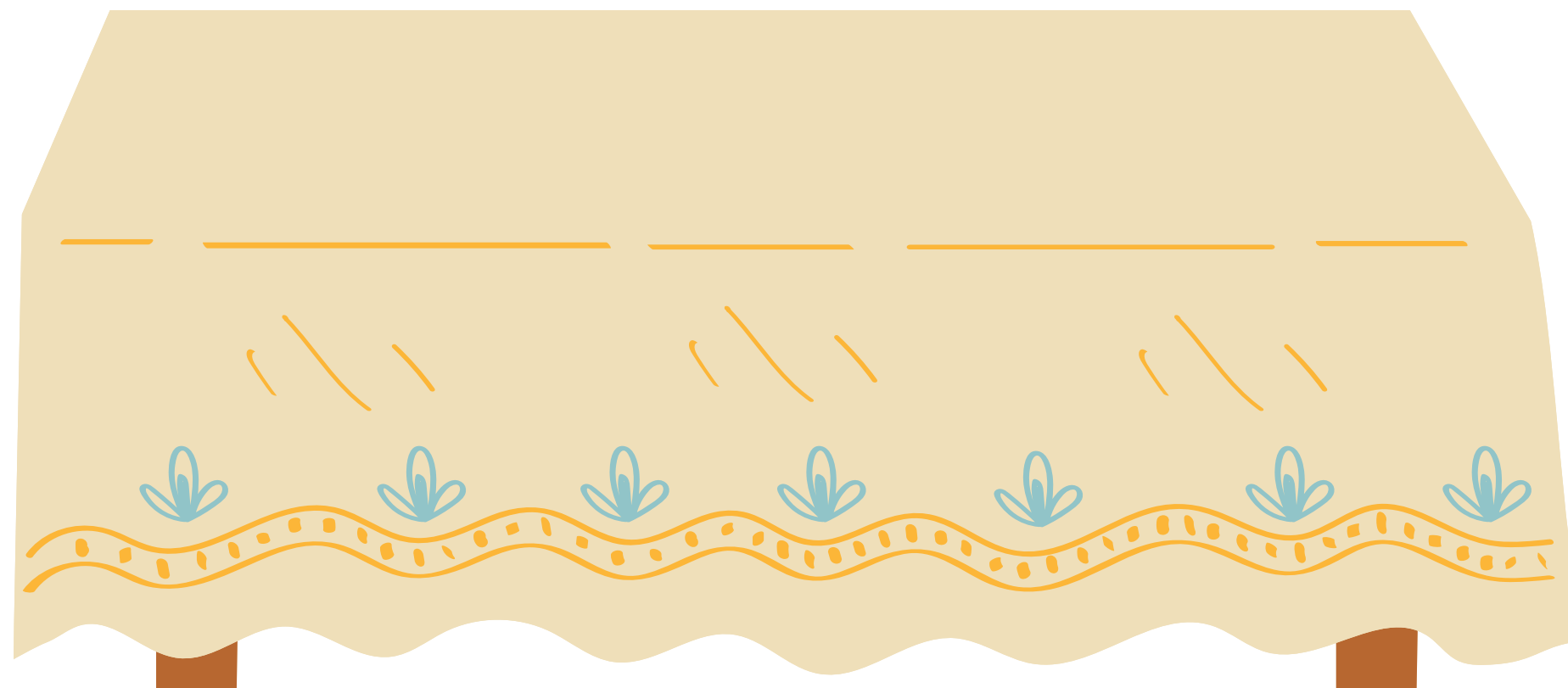
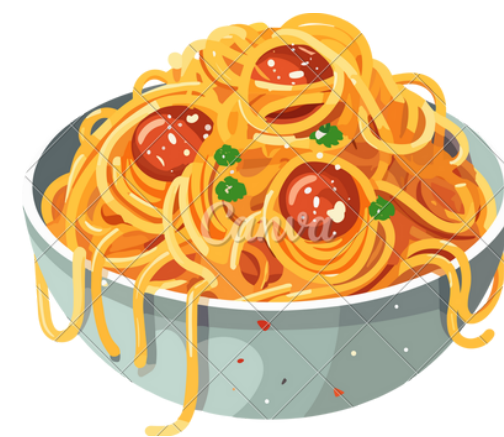
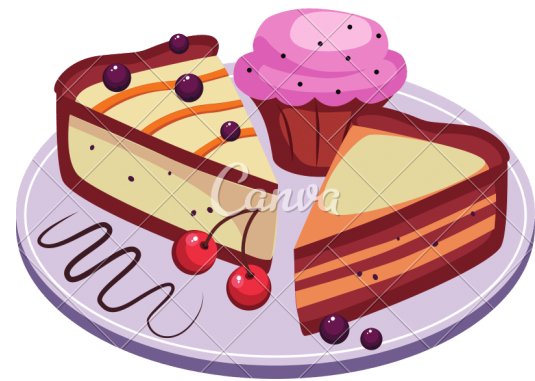
# A doua soluție:





Interdisciplinaritatea se remarcă prin faptul că, pentru a calcula numărul de soluții, folosim combinările. Totodată, problema poate fi utilă în domeniul culinar atunci când se pregătește meniul pentru un eveniment.

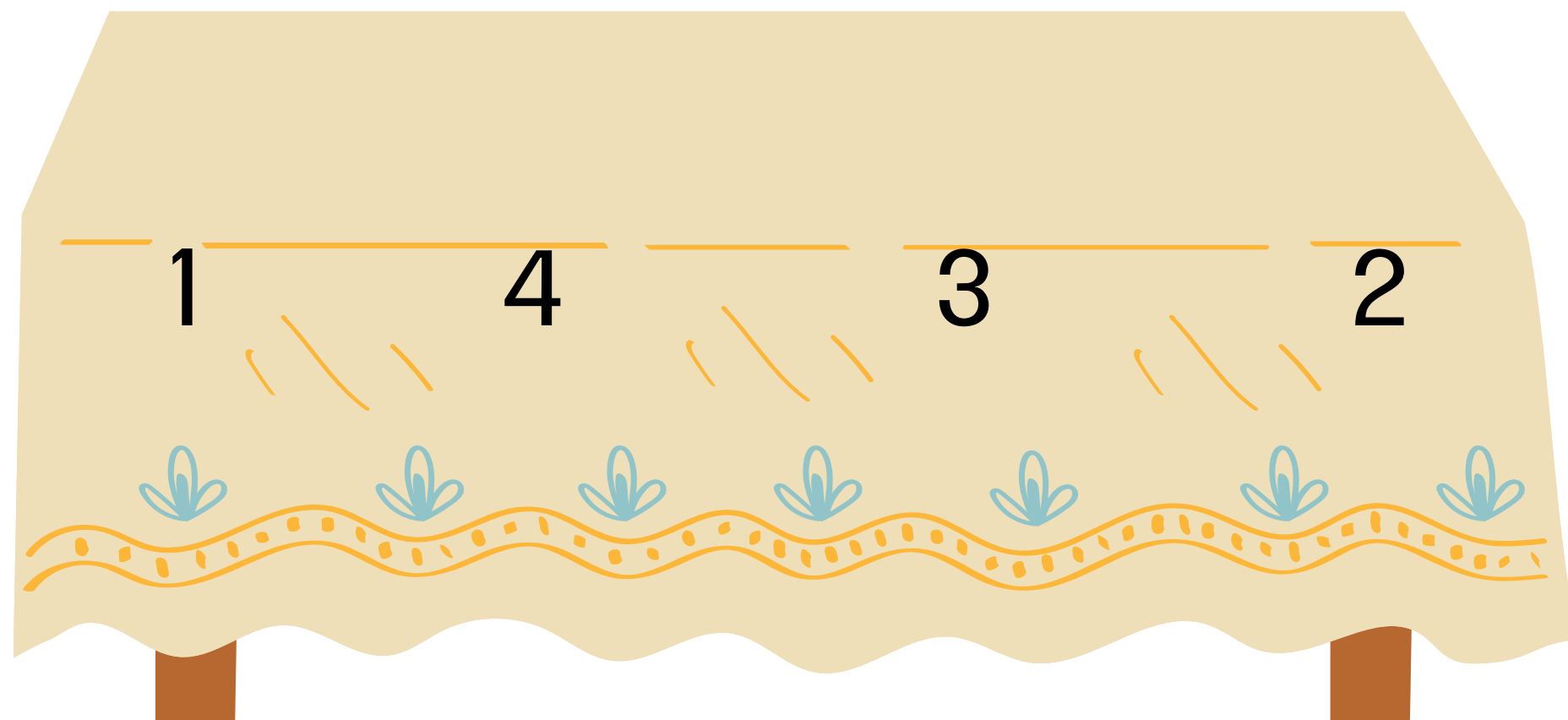
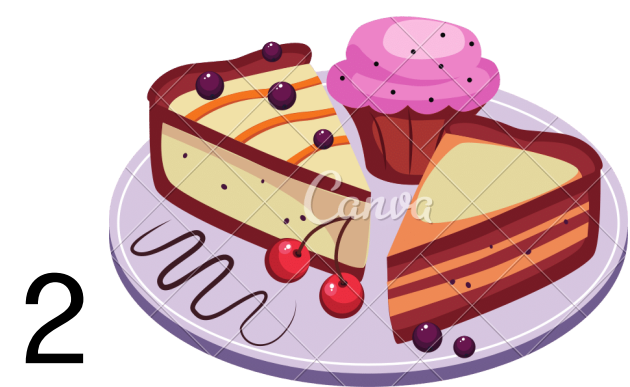
Pentru a calcula soluțiile, vom calcula ARANJAMENTE DE 4 LUATE CÂTE 2, deoarece ultimele două feluri de mâncare sunt stabilite



Am dat un exemplu de grupare câte 2 a elementelor.

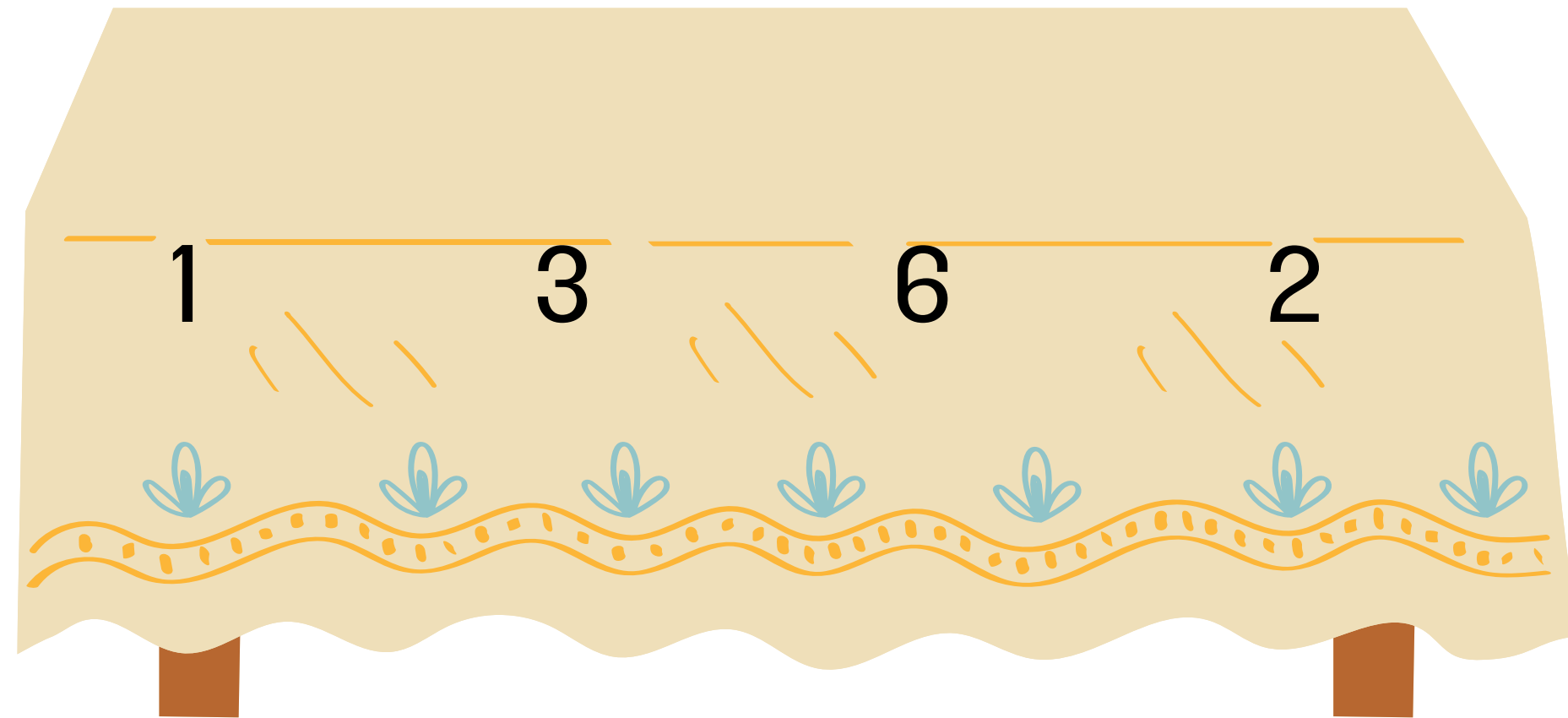
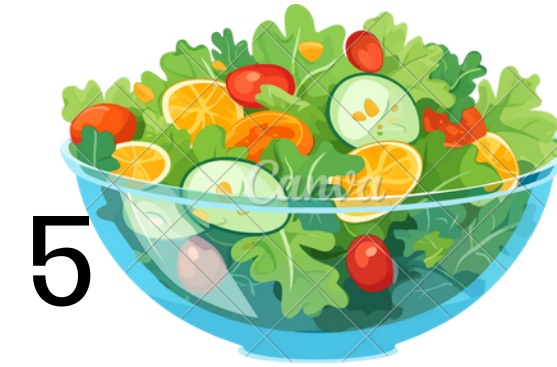
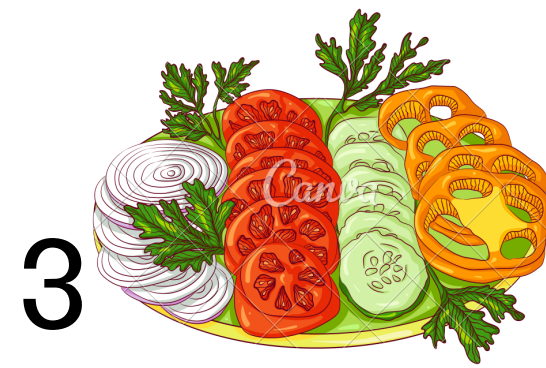
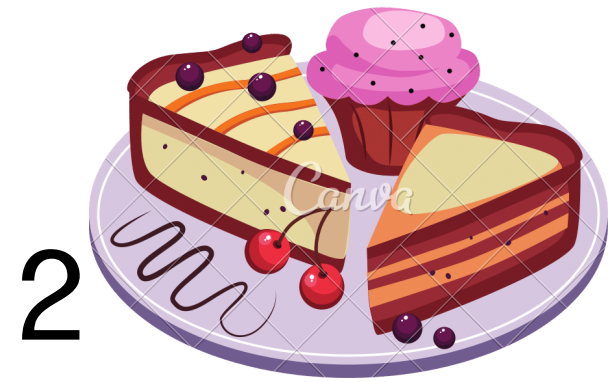
După efectuarea calculelor, se obțin în total 10 soluții.

Vom ilustra în continuare ultimele două soluții.





Ultima soluție obținută este :



3. Utilizând metoda backtracking se generează toate posibilitățile de a planta de-a lungul unei străzi cinci arbori din mulțimea {salcie, carpen, larice, fag, ulm}. Două soluții sunt diferite dacă ordinea arborilor diferă. Nu au voie să fie 2 arbori de același fel.





Salcie

Carpen

Larice

Fag

Ulm

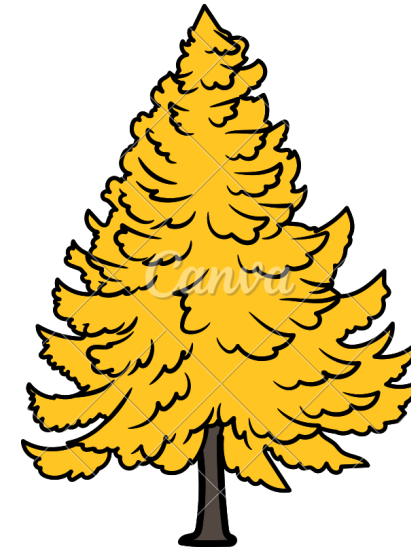
1



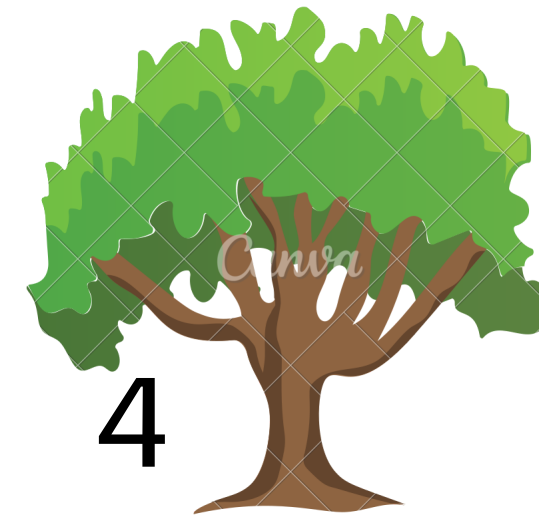
2



3



4



5



Vom genera primele 3 soluții  
Prima soluție:

1

2

3

4

5

# A doua soluție:

Salcie



1

Carpen



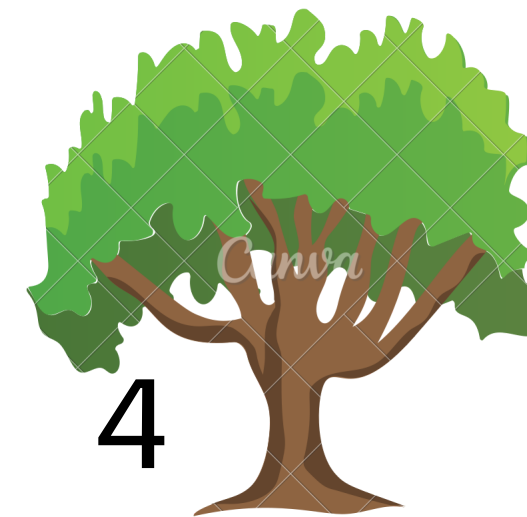
2

Larice



3

Fag



4

Ulm



5

1

2

3

5

4

A treia soluție:

Salcie



1

Carpen



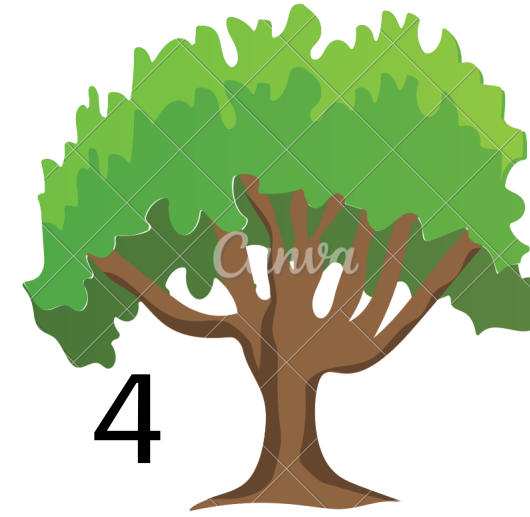
2

Larice



3

Fag



4

Ulm



5

1

2

3

5

4

# A treia soluție:

Salcie



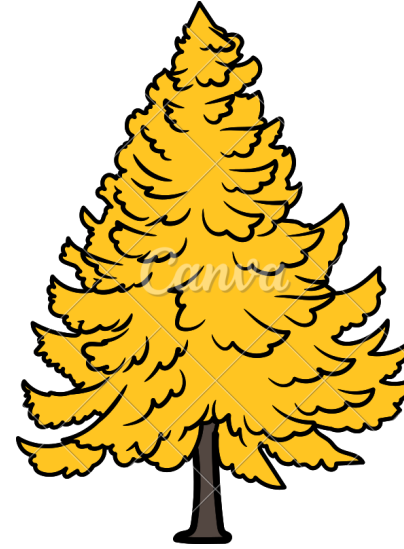
1

Carpen



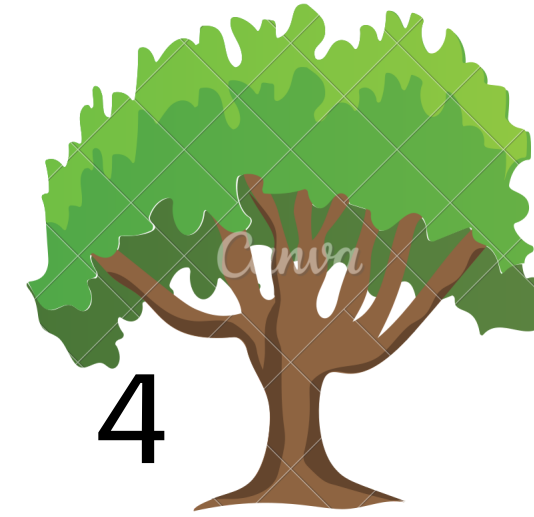
2

Larice



3

Fag



4

Ulm



5

1

2

4

3

5

# Algoritmul general

Metoda backtracking poate fi implementată iterativ sau recursiv. În ambele situații se folosește o structură de date de tip **stivă**. În cazul implementării iterative, stiva trebuie gestionată intern în algoritm – ceea ce poate duce la dificultăți în implementare. În cazul implementării recursive se folosește spațiu de memorie de tip stivă – STACK alocat programului; implementarea recursivă este de regulă mai scurtă și mai ușor de înțeles. Acest articol prezintă implementări recursive ale metodei.

Următorul subprogram recursiv prezintă algoritmul la modul general:

- la fiecare apel BACK(k) se generează valori pentru elementul  $x[k]$  al vectorului soluție;
- instrucțiunea Pentru modelează condițiile externe;
- subprogramul OK(k) verifică condițiile interne
- subprogramul Solutie(k) verifică dacă configurația curentă a vectorului soluție reprezintă o soluție finală
- subprogramul Afisare(k) tratează soluția curentă a problemei – de exemplu o afișează!

```
subprogram BACK(k)
```

```
┌ pentru fiecare element i din A[k] execută
```

```
|   x[k] ← A[i]
```

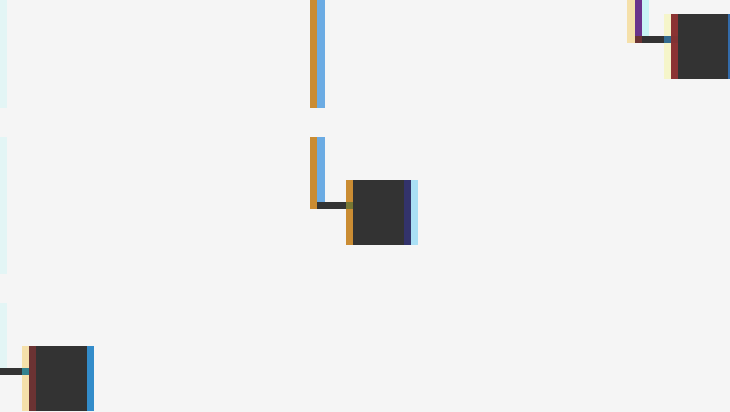
```
|   ┌ dacă OK(k) atunci
```

```
|   |   ┌ dacă Soluție(k) atunci
```

```
|   |   |   Afișare(k)
```

```
|   |   |   altfel
```

```
|   |   |   BACK(k+1)
```



```
sfârșit_subprogram
```