

Niech  $f$  słownikiem takim, że

$f[s]$  zwraca priorytet dla  $s$

$f[s] \leftarrow v$  dodaje  $s$  z priorytetem  $v$ , zastępując dotychczasowe  $s$

$\text{GetMin}(f, \text{set})$  zwraca element ze zbioru  $\text{set}$  mający najmniejszy priorytet nie usuwając go

$\text{GetSecondMin}(f, \text{set})$  zwraca element ze zbioru  $\text{set}$  mający drugi z kolei najmniejszy priorytet nie usuwając go

Algorytm rekurencyjny pierwszy najlepszy (RBFS) ma postać:

```
function RBFS(node, f_node, f_limit)
  if GoalTest(node) then return (node, 0)

  Initialize(f)
  switch Expand(node) do
    case  $\emptyset$ : return (failure,  $\infty$ )

    case {s}:
       $f[s] \leftarrow \max(g(s) + h(s), f\_node)$ 
      if  $f[s] > f\_limit$  then return (failure,  $f[s]$ )

      return RBFS(s,  $f[s]$ , f_limit)

    case succ:
      foreach  $s \in succ$  do  $f[s] \leftarrow \max(g(s) + h(s), f\_node)$ 

      loop do
        best  $\leftarrow$  GetMin(f, succ)
        if  $f[best] > f\_limit$  then return (failure,  $f[best]$ )

        alternative  $\leftarrow$  GetSecondMin(f, succ)
        result,  $f[best] \leftarrow$  RBFS(best,  $f[best]$ ,  $\min(f\_limit, f[alternative])$ )
        if result  $\neq$  failure then return (result, 0)
      end
    endsw
  end
end

RBFS(Initial_node,  $h(\text{Initial\_node})$ ,  $\infty$ )
```

## Zadanie 1

Dla układanki 5 puzzli zaprezentować działanie algorytmu rekurencyjnego przeszukiwania pierwszy najlepszy.

\_24 => 123  
153     45\_

Heurystyka to koszt rozwiązania przy założeniu że wiele puzzli może stać na tym samym polu (czyli min. suma przesunięć potrzebnych do rozwiązania bez uwzględnienia tego, że puzzle przeszkadzają sobie nawzajem). W podanym stanie startowym  $1+0+1+3+0 = 5$ .

Dla każdego stanu podaj wartość  $f$ , dla stanów dla których algorytm został rekurencyjnie wywołany podaj dodatkowo wartość  $f\_limit$ .

## Zadanie 2

Odpowiedz na następujące pytania:

Czy algorytm RBFS może wielokrotnie rozwijać ten sam wierzchołek w danej ścieżce?

Czy algorytm RBFS się pętli?  
Czy algorytm RBFS znajduje (optymalne) rozwiązanie?  
Jaką RBFS ma złożoność pamięciową?

### Zadanie 3

Funkcja oceny stanu dana jest wzorem:  $f(n) = (2 - w)g(n) + wh(n)$ , gdzie:

$g(n)$  - rzeczywisty koszt dojścia do stanu  $n$

$h(n)$  - wartość funkcji heurystycznej w stanie  $n$

$w$  - parametr

Zakładamy, że uruchamiamy przeszukiwanie pierwszy najlepszy z funkcja oceny  $f$ .

Jakie algorytmy dostajemy dla  $w = 0, 1, 2$ ?

Dla jakich wartości parametru  $w$  przeszukiwanie pierwszy najlepszy pozostaje optymalne?

### Zadanie 4 *Przeszukiwanie iteracyjnie wydłużane*

Przeszukiwanie iteracyjnie wydłużane jest iteracyjnym algorytmem analogicznym do strategii jednolitego kosztu. Polega na przeszukiwaniu włąb z ograniczeniem maksymalnego kosztu ścieżki (jeśli koszt dojścia do wierzchołka przekroczy limit, wierzchołek jest odrzucany). Jeżeli przy danym ograniczeniu stan docelowy nie zostanie znaleziony, określany jest nowy limit równy najmniejszemu kosztowi ścieżki prowadzącej do jednego z odrzuconych wierzchołków.

- Pokaż, że algorytm znajduje optymalne rozwiązanie.
- Rozważmy drzewo o stopniu rozgałęzienia  $b$ , rozwiązaniu na głębokości  $d$  i jednakowym koszcie wszystkich krawędzi. Ile iteracji wymaga algorytm, ile wierzchołków w sumie odwiedzi?
- Rozważmy drzewo o stopniu rozgałęzienia  $b$ , rozwiązaniu na głębokości  $d$ . Koszty krawędzi należą do  $[0, 1]$  i są większe od  $\varepsilon > 0$ . Ile iteracji wymaga algorytm, ile wierzchołków w sumie odwiedzi?