



Prezentacja projektu z Algorytmów i Struktur Danych II



Skład naszej grupy



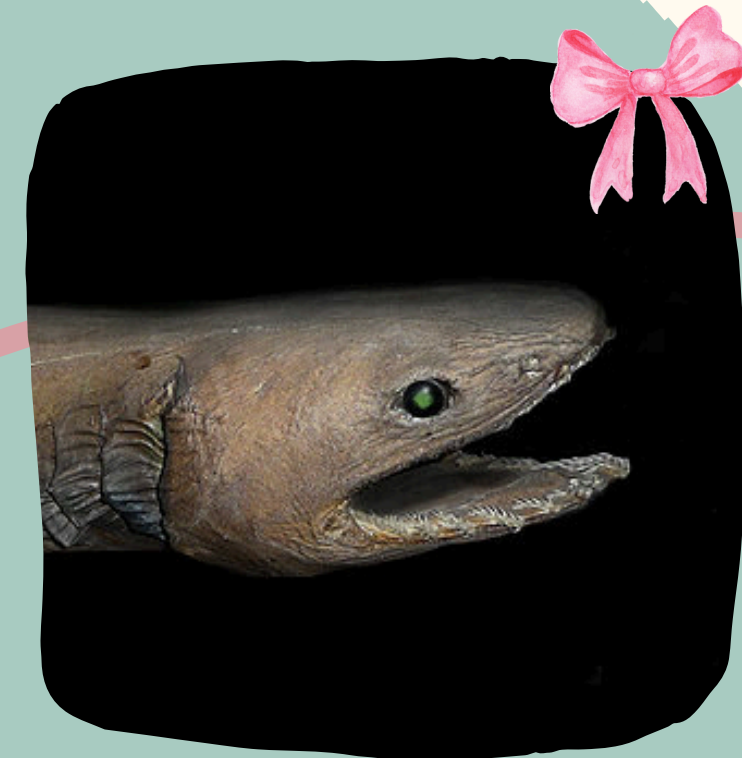
**Mikołaj
Blangiewicz**

Kierownik grupy,
programista



**Natan
Warelich**

programista
mentalne wsparcie



**Nadia
Hermann**

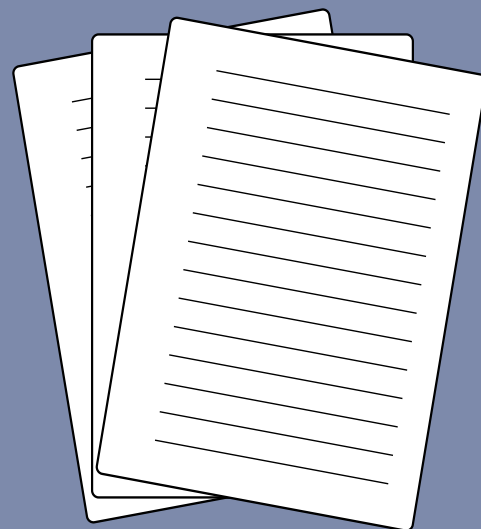
programistka,
prezentacja

Omwiane problemy

Problem I



Problem II



Problem III





Problem I

I część zadania

II część zadania

III część zadania

**(wszystkie problemy będą docelowo rozszerzone w
innych slajdach)**



Problem II

I część zadania

II część zadania

III część zadania

Omówienie problemu

Problem. Zapisać opowieść-melodię w maszynie Informatyka, zamieniając wcześniej „poli” na „boli” oraz próbując oszczędzić wykorzystane miejsce. Znaleźć rozwiązanie problemu ewentualnej zamiany innych fragmentów opowieści-melodii, który niepokoi Heretyka oraz Informatyka.

Czyli ogólnie rzecz ujmując musimy:

- 1. Znaleźć wzorzec w tekście**
- 2. Poddać całość kompresji**

Zastosowane algorytmy

- **algorytm kompresji Huffmana**
- **algorytm wyszukiwania wzorca Knutha Morissa Pratta**

wejście

```
≡ song.txt ×  
1 Gdzies jest, lecz nie wiadomo gdzie  
2 Swiat, w ktorym basn ta dzieje sie,  
3 Malenka pszczolka mieszka w nim  
4 Co wiesc chce wsrod owadow prym.  
5 Te pszczolke, ktora tu widzicie, zowia Ania,  
6 Wszyscy Anie znaja i kochaja.  
7 Ania fruwa tu i tam  
8 Swiat swoj pokazujac nam.  
9 Dzis spotka Was malenka, zwinna pszczolka Ania,  
10 Smiala, sprytna, rezolutna Ania  
11 Mala przyjaciolka Ania  
12 Aniu, (Aniu) Aniu, (Aniu)  
13 Aniu coz zobaczymy dzis?  
14 Mala, rezolutna Ania,  
15 Aniu, (Aniu) baju, (baju)  
16 Aniu coz zobaczymy dzis
```

```
≡ words.txt ×  
1 Ania Maja  
2 Anie Maje  
3 Aniu Maju
```


wyjście

```
w: 10000
y: 111011
z: 0101
Alright mr Heretic, we have sucessfully made compression, what do we do now?
type "1" if you want to see what did you do.
type "2" if you want to decrypt code.
type "3" if you want to use 2d printer to save results. (it ends program and input results into file)
1
110110000111010010100011111011100011001011110111000100110010011001101111010001010101100001000111110011100000001101111010
1111100000111110110100011011110100101000111110110010100001100000001101010011001001110000011001110100111111101111101100
0000111101101101111100000010110100110101111101001010001111100010111100111110000011111010010110011101010100110111100000100
111101011110001111000101100010101111110011000111101011000000001111101110001010011110101110000011000010001000001100111011
000111111011100000001111101110010001011100010100010110001111100111000011100110111111111010011111110000101111010111111
000001111000111011111101100000010000011001010001100111100111100011110001011000101011111100110001111111010010011001110100
1111111011110101101001100110111000000011110100101000110001000111110100100110101111110000000110101111010101001010110010
110011101100111100010111101111100100011110110111101010100101111001101010000110100101011000101100111111110001010001011
010010101010000011001110101010010101011010001110110111100111000010101101001100110110001011010011010000011001010000110000
00011010100101111100100001111100100111100011111001111010101100110010101100010110000110100000010000011001010001000101000
11110001111001100011111101001001111010111101100110111100011000001010011011110000010011110110010011010110000000100001000
011010111100011110001011000101011111100110001111010111101010100101011001011001010000100000000110100110101100100111110011
0001110111111011010010000110110010011110111111001011111001101001101001000011010111101010100101011100111010101001101010
111100011101110101111011001010110001000111111001100011110101111010101001010111001110101010010100111001001111000011101010
100101001111000000111101010100101001110010011110000111010101001010011110000011001110101010010100110111000111111010101101
01111111101101101100010101111011000001110110111110100101000111100010001111110011101010100110101100100111101111110010111
111001101001101001000011010111101010100101011001011001110101010010100111001001111000011101010100101001111000000111101101
101001010011100100111100001110110110100101001111000001100111010101001010011011100011111101010110110001011000101100010
1011110110000011101101111101001010001111000100011111100111010101001101011001001111011111110010111
```

type "1" if you want to see what did you do.
type "2" if you want to decrypt code.
type "3" if you want to use 2d printer to save results. (it ends program and input results into file)

2

Gdzies jest, lecz nie wiadomo gdzie
Swiat, w ktorym basn ta dzieje sie,
Malenka pszczolka mieszka w nim
Co wiesc chce wsrod owadow prym.
Te pszczolke, ktora tu widzicie, zowia Maja,
Wszyscy Maje znaja i kochaja.
Maja fruwa tu i tam
Swiat swoj pokazujac nam.
Dzis spotka Was malenka, zwinna pszczolka Maja,
Smiala, sprytna, rezolutna Maja
Mala przyjaciolka Maja
Maju, (Maju) Maju, (Maju)
Maju coz zobaczymy dzis?
Mala, rezolutna Maja,
Maju, (Maju) baju, (baju)
Maju coz zobaczymy dzis

type "1" if you want to see what did you do.
type "2" if you want to decrypt code.
type "3" if you want to use 2d printer to save results. (it ends program and input results into file)

3

skrajne przypadki

a -> b
c -> d
b -> c

wejście:
aaa

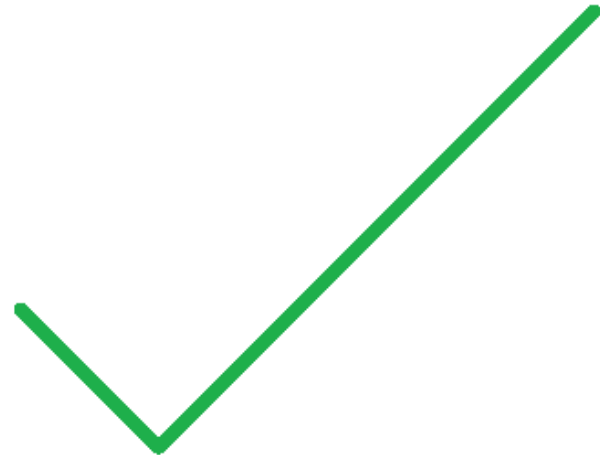
otrzymane wyjście:
bbb

prawidłowe wyjście:
ddd

rower -> kajak
kaj -> maj

wejście:
rower

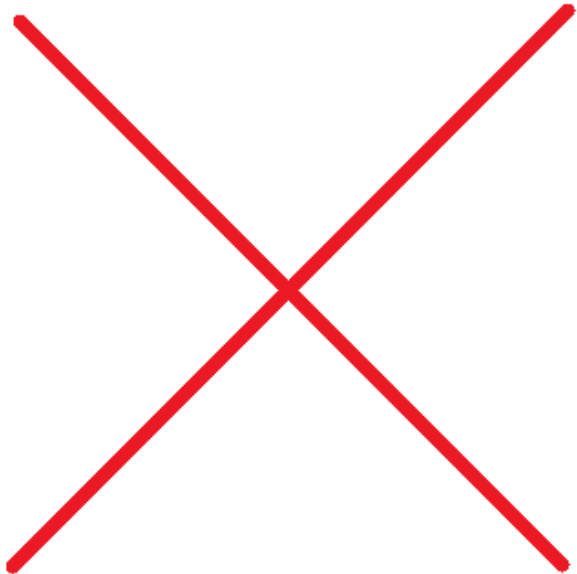
wyjście:
majak



kaj -> maj
rower -> kajak

wejście:
rower

wyjście:
kajak



Rozwiązanie?

```
Filtering::sortByLength( &: words);  
for(i = 0; i < n; i++){  
    kmp.pattern( &: song, pattern: words[i].first, fixedword: words[i].second);  
}
```

```
Filtering::sortByReplacement( &: words);  
for(i = 0; i < n; i++){  
    kmp.pattern( &: song, pattern: words[i].first, fixedword: words[i].second);  
}
```

```
Filtering::sortByReplacement2( &: words);  
for(i = 0; i < n; i++){  
    kmp.pattern( &: song, pattern: words[i].first, fixedword: words[i].second);  
}
```

złożoność

czasowa:

KMP: $O(k \cdot (n+m))$

Huffmana: $O(n \cdot \log n)$

Uproszczona ogólna:

$O(k \cdot (n + m) + n \cdot \log n)$

pamięciowa:

Odczyt plików:

$O(n)$

KMP: $O(n)$

Huffman: $O(n)$



Problem III

I problem

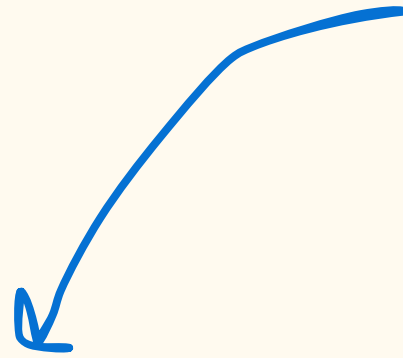
**Ustalić grafik płaszczków z
uwzględnieniem wielu warunków**

II problem

**Ustalenie najkorzystniejszej ścieżki
dla strażników, aby ilość przestępstw
była jak najmniejsza**



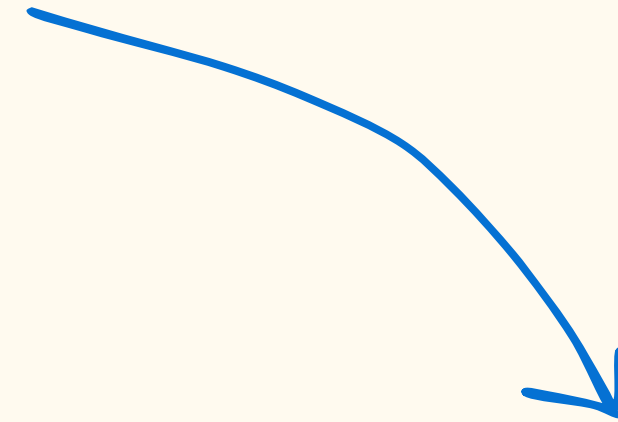
Interpretacja zadania



Grafik strażników



Ustalenie ścieżki



Dni wolne

Budowa rozwiązania

```
▼ folder txt
  ≡≡≡ p3_path1.txt
  ≡≡≡ p3_vacation1.txt
  📄 CMakeLists.txt
  📄 guards.cpp
  📄 guards.h
  📄 problem3.cpp
  📄 program.exe
  📄 route.cpp
  📄 route.h
```



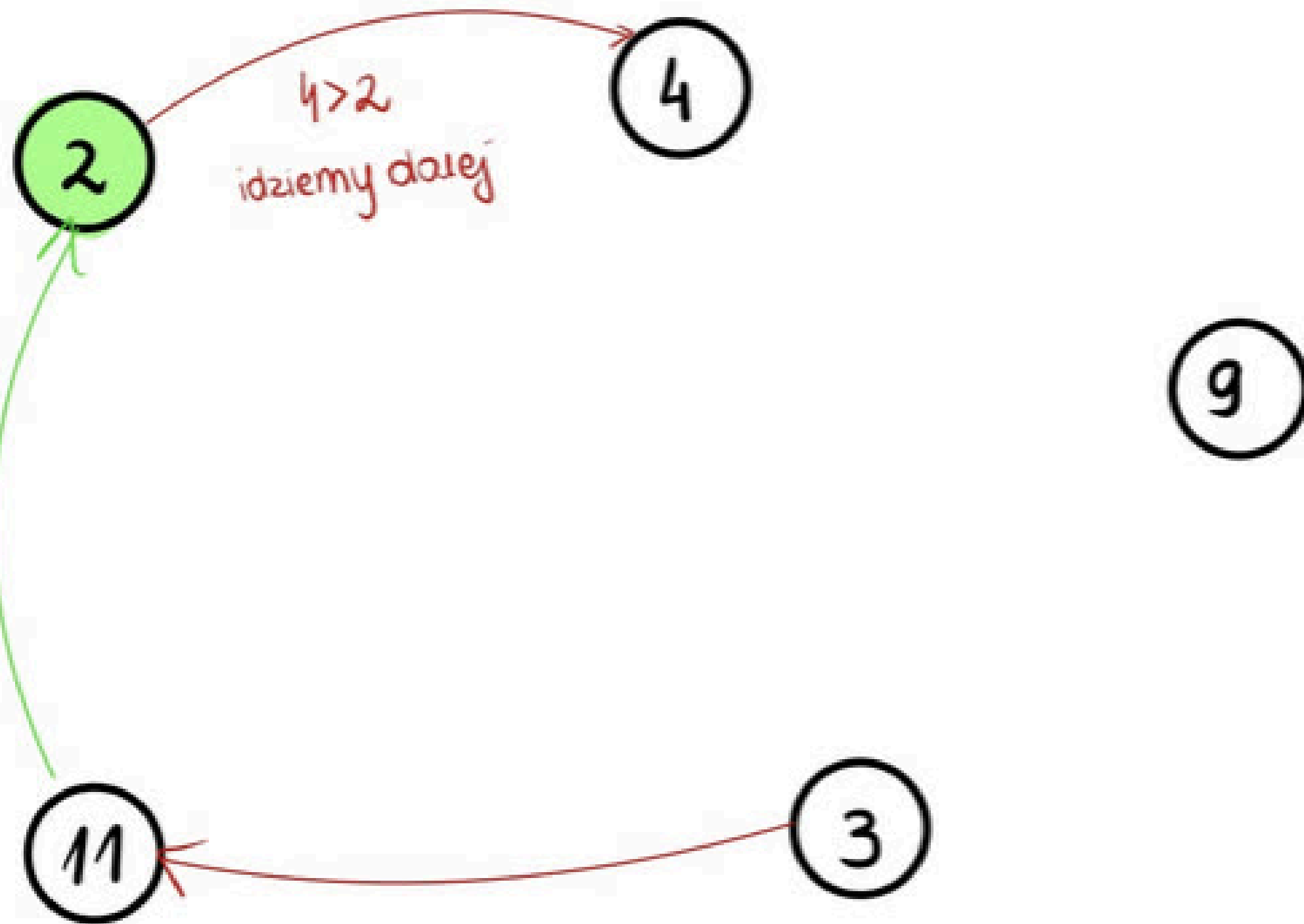

Zastosowane funkcje i struktury

```
Guards::Guards(const std::unordered_map<int, int>& initialEnergies, const std::unordered_map<int, int>& vacationMap) : energyMap(initialEnergies), vacationMap(vacations) {
    for (const auto& entry : const pair<...>& : initialEnergies) {
        guardQueue.push(x: { .id: entry.first, .energy: entry.second});
        availableGuards.insert(x: entry.first);
    }
}
```

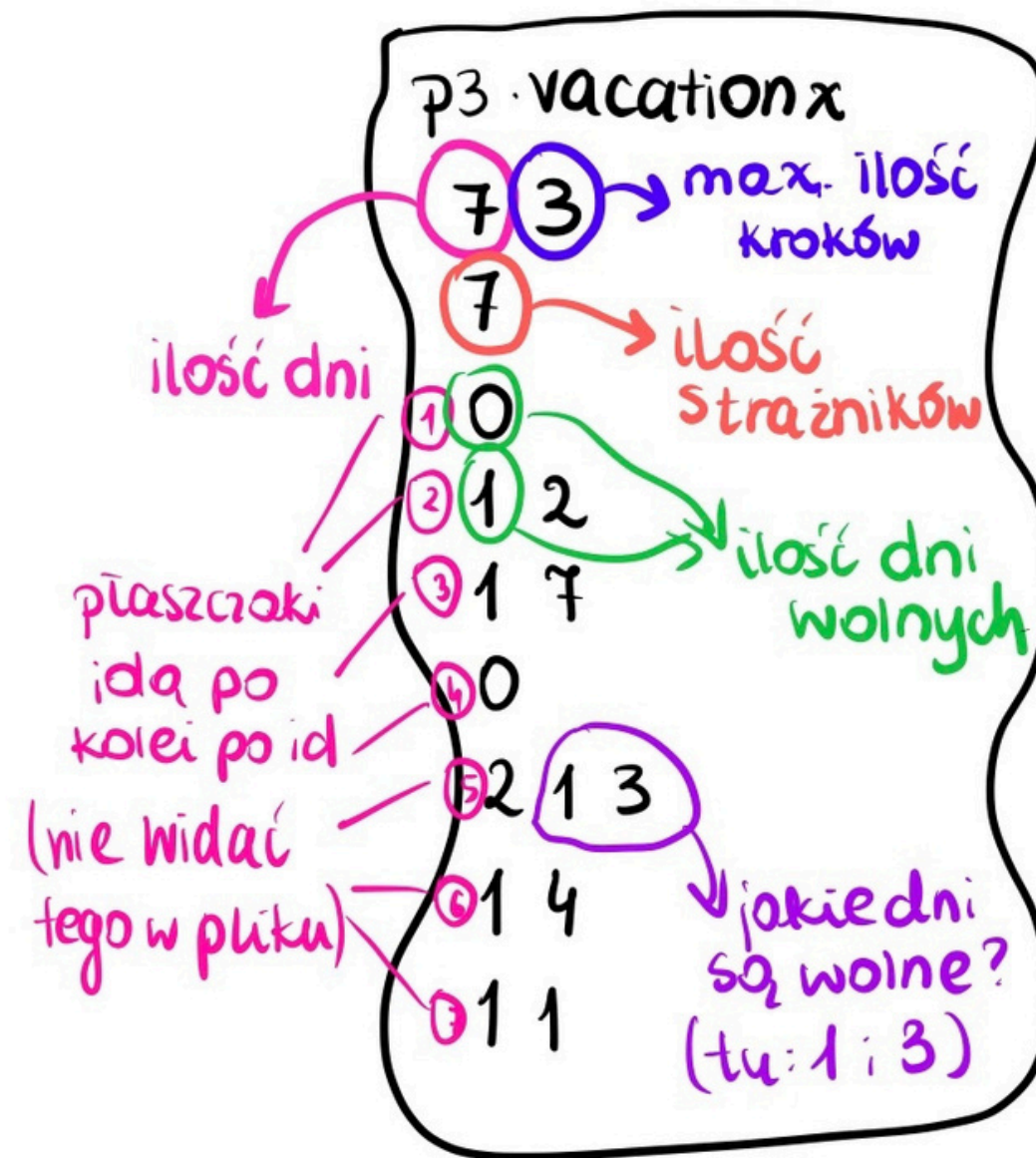
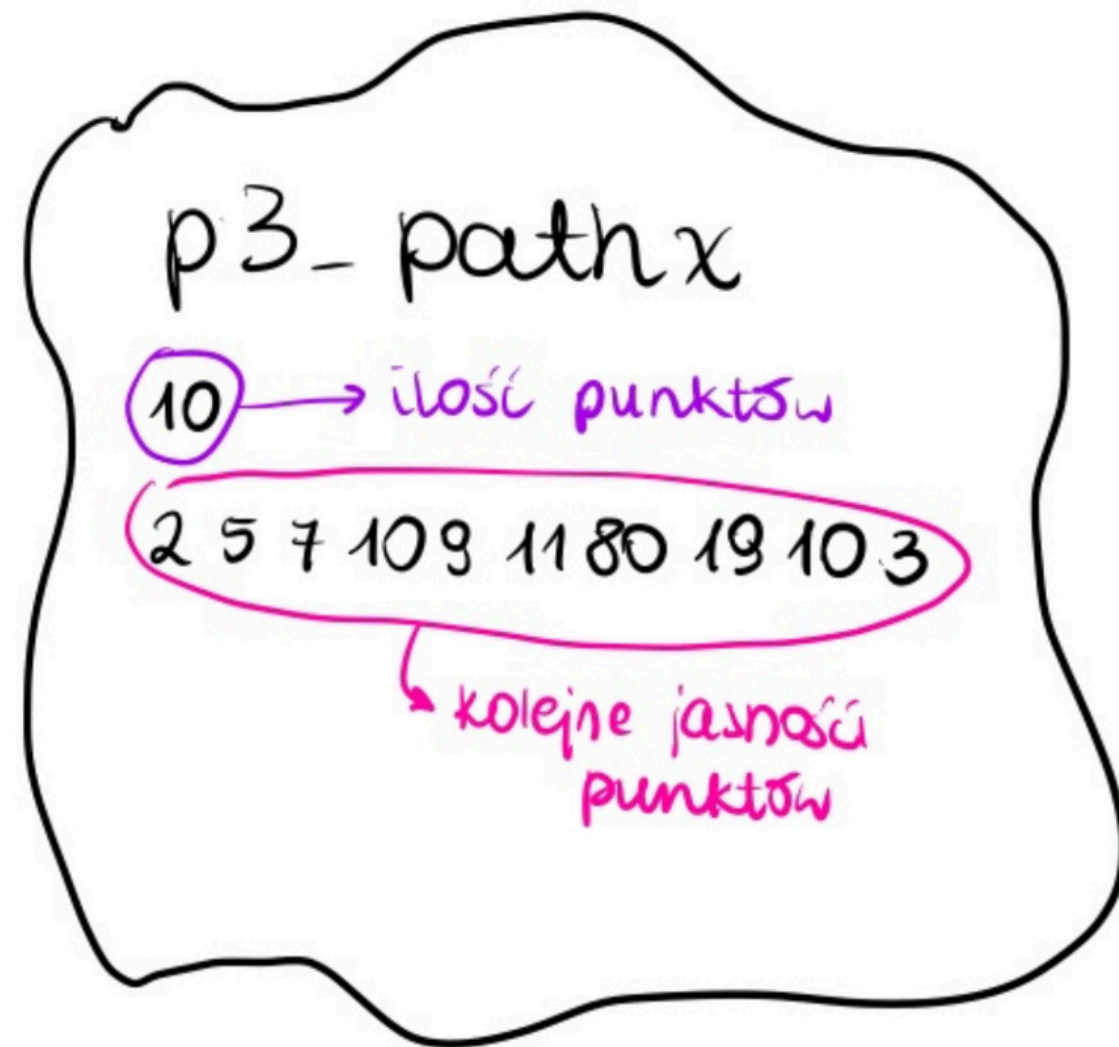
```
Guard Guards::selectGuard(int day) {
    std::priority_queue<Guard> tempQueue;
    Guard selectedGuard = { .id: 0, .energy: 0};

    for (auto it : iterator<...> = restMap.begin(); it != restMap.end();) {
        if (it->second <= day) {
            availableGuards.insert(x: it->first);
            it = restMap.erase(position: it);
        } else {
            ++it;
        }
    }
}
```

max_steps = 2



Omówienie wejść



Omówienie wyjścia

```
Test number: 1
```

```
Guard 1 initial energy: 10
```

```
Guard 2 initial energy: 1
```

```
Guard 3 initial energy: 10
```

```
Guard 4 initial energy: 9
```

```
Guard 5 initial energy: 7
```

```
Guard 6 initial energy: 3
```

```
Guard 7 initial energy: 8
```

```
Day 1: Guard 3
```

```
Day 2: Guard 1
```

```
Day 3: Guard 7
```

```
Day 4: Guard 4
```

```
Day 5: Guard 5
```

```
Day 6: Guard 6
```

```
Day 7: Guard 2
```

```
Guards' path goes with 9 stops and 3 listens to the melody.
```

Przypadki normalne

**Przypadki skrajne dla grafiku:
jeden płaszczak ma wolne przez
cały tydzień, każdy płaszczak ma
wolne tego samego dnia**

**Przypadki skrajne dla
ścieżki: jasności kolejne,
które są o 1 większe niż
poprzednie**

**Przeprowadzone
testy**



Dziękujemy!

Zachęcamy do zadawania pytań