



РТУ МИРЭА. ИИТ ИКБО-09-22

Сортировка перестановками (PermSort)

Кузнецов Ярослав Андреевич

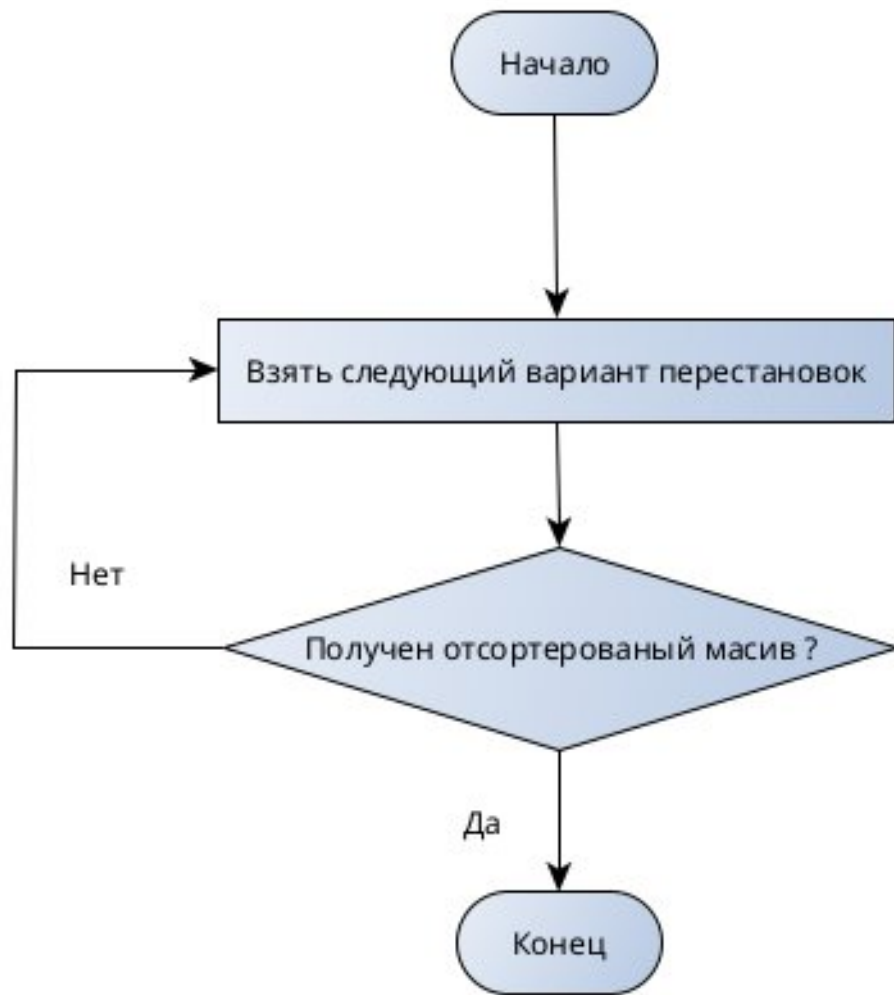
г.Москва 2022г.

История



- Алгоритм разработан учёным Национального Института Стандартов и Технологий (США) Паулем Е. Блеком в 2001г.
- Сортировка создана чисто в академических целях и не несёт практического смысла.

Алгоритм



- Взгляд со стороны комбинаторики.
- Алгоритм этой сортировки предельно прост. Нужно перебирать все возможные варианты перестановок пока не попадётся отсортированная.
- В лучшем случае сложность алгоритма $O(n)$, а в худшем $O(n * n!)$

Реализация на C++

```
std::vector<int_fast32_t> PermSort (std::vector<int_fast32_t> &data, std::vector<int_fast32_t> variants){  
    std::vector<int_fast32_t> temp_vector = data;  
    if (variants.size() == 1){  
        temp_vector.push_back(variants[0]);  
        if (is_sorted(temp_vector)){  
            return temp_vector;  
        }  
        return ZERO_DATA;  
    }  
    std::vector<int_fast32_t> result;  
    std::vector<int_fast32_t> vr2;  
    for (int_fast32_t i = 0; i < variants.size(); i++){  
        temp_vector = data;  
        vr2 = variants;  
        temp_vector.push_back(pop_item(i, vr2));  
        result = PermSort(temp_vector, vr2);  
        if (result != ZERO_DATA){  
            return result;  
        }  
    }  
    return ZERO_DATA;  
}
```

Реализация на C++

```
//  
// Created by yaroslav_admin on 02.10.22.  
//
```

```
#include <iostream>  
#include <vector>
```

```
std::vector<int_fast32_t> ZERO_DATA = {};
```

```
int main(){
```

```
    std::vector<int_fast32_t>  
    data = {3, 2, 5, -5, 8, 9, -11, 24, -37, 0};  
    std::vector<int_fast32_t> sorted =  
    PermSort(ZERO_DATA, data);  
    print_vector(sorted);
```

```
    return 0;  
}
```

```
void print_vector (std::vector<int_fast32_t> &data) {  
    for (int_fast32_t i : data){  
        std::cout << std::to_string((int)i) << " ";  
    }  
    std::cout << std::endl;  
}
```

```
int_fast32_t pop_item(int_fast32_t index, std::vector<int_fast32_t> &data){  
    int_fast32_t result = data[index];  
    data.erase(data.begin() + index);  
    return result;  
}
```

```
bool is_sorted(std::vector<int_fast32_t> &data){  
    for(int_fast32_t i = 0; i < data.size() - 1; i++){  
        if (data[i + 1] < data[i]){  
            return false;  
        }  
    }  
    return true;  
}
```

Реализация на разных ЯП.

```
//  
// Created by yaroslav_admin on 01.10.22.  
//
```

```
#include <iostream>  
#include <vector>  
std::vector<int> ZERO_DATA = {};
```

```
void print_vector(std::vector<int> &data) {  
    for (int i : data) {  
        std::cout << i << « «<br>    }  
    std::cout << std::endl;  
}
```

```
int pop_item(int index, std::vector<int> &data) {  
    int result = data[index];  
    data.erase(data.begin() + index);  
    return result;  
}
```

```
bool is_sorted(std::vector<int> &data) {  
    for (unsigned int i = 0; i < data.size() - 1; i++) {  
        if (data[i + 1] < data[i]) {  
            return false;  
        }  
    }  
    return true;  
}
```

```
std::vector<int> PermSort(std::vector<int> &data, std::vector<int> variants) {  
    std::vector<int> temp_vector = data;  
    if (variants.size() == 1) {  
        temp_vector.push_back(variants[0]);  
        if (is_sorted(temp_vector)) {  
            return temp_vector;  
        }  
        return ZERO_DATA;  
    }  
    std::vector<int> result;  
    std::vector<int> vr2;  
    for (int i = 0; i < variants.size(); i++) {  
        temp_vector = data;  
        vr2 = variants;  
        temp_vector.push_back(pop_item(i, vr2));  
        result = PermSort(temp_vector, vr2);  
        if (result != ZERO_DATA) {  
            return result;  
        }  
    }  
    return ZERO_DATA;  
}
```

```
int main() {  
    std::vector<int> data = {3, 2, 5, -5, 8, 9, -11};  
    std::vector<int> sorted = PermSort(ZERO_DATA, data);  
    print_vector(sorted);  
    return 0;  
}
```

```
ZERO_DATA = list()
```

```
def perm_sort(data: list, variants: list) -> list:  
    if len(variants) == 1:  
        if is_sorted(data + variants):  
            return data + variants  
        return ZERO_DATA  
    for i in variants:  
        temp = variants.copy()  
        temp.remove(i)  
        result = perm_sort(data + [i], temp)  
        if result != ZERO_DATA:  
            return result  
    return ZERO_DATA
```

```
def is_sorted(data: list) -> bool:  
    for i in range(len(data) - 1):  
        if data[i + 1] < data[i]:  
            return False  
    return True
```

```
def main():  
    data = [3, 2, 5, -5, 8, 9, -11]  
    data_sorted = perm_sort([], data)  
    print(*data_sorted)
```

```
if __name__ == '__main__':  
    main()
```



C++ (Linux)



C++



Python

Производительность:

	3, 2, 5, -5, 8, 9, -11	3, 2, 5, -5, 8, 9, -11, 24, -37	3, 2, 5, -5, 8, 9, -11, 24, -37, 0	3, 2, 5, -5, 8, 9, -11, 24, -37, 0, 444
Python3	0m0,084s	0m0,638s	0m5,228s	0m59,336s
C++ (Стандартный код)	0m0,020s	0m0,891s	0m8,039s	1m27,738s
C++ (С оптимизациями Linux)	0m0,003s	0m0,108s	0m0,847s	0m8,837s

Характеристики тестовой машины:

- CPU: Intel i5 750 (4) @ 2.661GHz
- GPU: AMD ATI Radeon RX 580
- Memory: 16 GB
- OS: Fedora Linux 36 (Workstation Edition) x86_64
- Kernel: 5.19.10-200.fc36.x86_64



Скрипт для замеров и компиляции

ИСТОЧНИКИ:

- Шагбазян Д.В. Алгоритмы сортировки. Анализ, реализация, применение: учебное пособие / Д.В. Шагбазян, А.А. Штанюк, Е.В. Малкина. – Нижний Новгород: Нижегородский госуниверситет, 2019. – 42 с.
- permutation sort // National Institute of Standards and Technology a : [сайт]. – 2021. – URL: <https://xlinux.nist.gov/dads/HTML/permutationSort.html> (дата обращения: 01.10.2022).
- National Institute of Standards and Technology официальный сайт. – США – Обновляется в течение суток.. –URL: <https://www.nist.gov/> (дата обращения 1.10.2022).