# Отчёт №5: «Жадные алгоритмы».

Жадный алгоритм – алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.

Известно, что если структура задачи задаётся матроидом, тогда применение жадного алгоритма выдаст глобальный оптимум.

Матроид — классификация подмножеств некоторого множества, представляющая собой обобщение идеи независимости элементов, аналогично независимости элементов линейного пространства, на произвольное множество.

Если глобальная оптимальность алгоритма имеет место практически всегда, его обычно предпочитают другим методам оптимизации, таким как динамическое программирование.

Общего критерия оценки применимости жадного алгоритма для решения конкретной задачи не существует, однако для задач, решаемых жадными алгоритмами, характерны две особенности: принцип жадного выбора и оптимальности для подзадач.

Говорят, что к оптимизационной задачи применим принцип жадного выбора, если последовательность локально оптимальных выборов даёт глобально оптимальное решение.

Говорят, что задача обладает свойством оптимальности для подзадач, если оптимальное решение задачи содержит в себе оптимальные решения для всех её подзадач.

К примеру, [алгоритм Дейкстры](http://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D1%8B) нахождения кратчайшего пути в графе вполне себе жадный, потому что мы на каждом шагу ищем вершину с наименьшим весом, в которой мы еще не бывали, после чего обновляем значения других вершин. При этом можно доказать, что кратчайшие пути, найденные в

вершинах, являются оптимальными.

К слову, [алгоритм Флойда](http://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A4%D0%BB%D0%BE%D0%B9%D0%B4%D0%B0_%E2%80%94_%D0%A3%D0%BE%D1%80%D1%88%D0%B5%D0%BB%D0%BB%D0%B0), который тоже ищет кратчайшие пути в графе (правда, между всеми вершинами), не является примером жадного алгоритма.

# Задача о расписании

Пусть программисту-фрилансеру Васе дано n заданий. У каждого задания

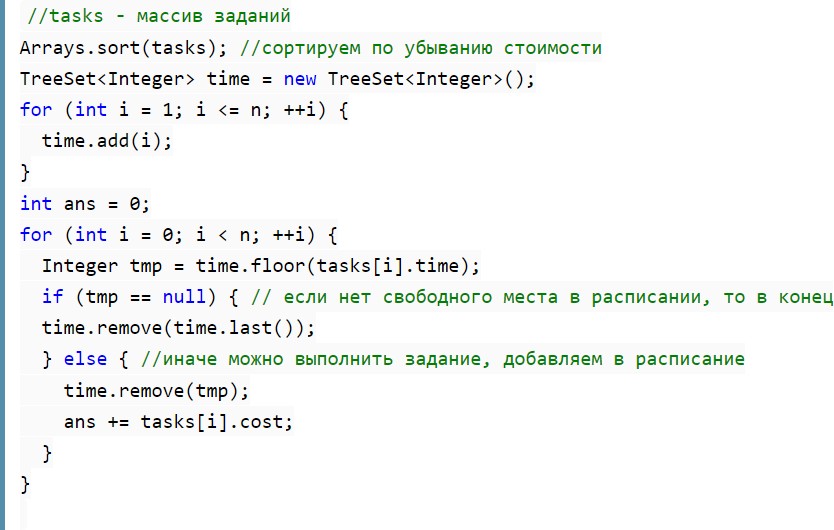
известен свой дедлайн, а также его стоимость (то есть если он не выполняет это задание, то он теряет столько-то денег). Вася настолько крут, что за один день может сделать одно задание. Выполнение задания можно начать с момента 0. Нужно максимизировать прибыль.

Классический пример применения жадины: Васе выгодно делать самые

«дорогие задания», а наименее дорогие можно и не выполнять — тогда прибыль будет максимальна. Возникает вопрос: каким образом

распределить задания? Будем перебирать задания в порядке убывания стоимости и заполнять расписание следующим образом: если для заказа

есть еще хотя бы одно свободное место в расписании раньше его дедлайна, то поставим его на самое последнее из таких мест, в противном случае в срок мы его не можем выполнить, значит поставим в конец из свободных мест.



Пример жадного алгоритма из лабораторной работы

vector<int> jaba(vector<int>& arr, int kyp)

{

sort(arr.begin(), arr.end());

int i = arr.size() - 1;

vector<int> razmen;

while (i >= 0 || kyp != 0)

{

if (kyp - arr[i] >= 0)

{

razmen.push\_back(arr[i]);

kyp = kyp - arr[i];

}

else

{

i--;

}

}

return razmen;

}

Источники:

* <https://ru.wikipedia.org/wiki/Жадный_алгоритм>
* https://habr.com/ru/post/120343/