1. **Можно ли с помощью алгоритма с возвратом решить любую *NP*-полную задачу?**

При большом количестве элементов, алгоритм полного перебора может работать очень долго, а для некоторых задач, перебор всех путей в графе, необходимо использовать именно такой алгоритм, из-за слишком большого времени, на данный момент времени мы не может решить любую задачу за полиномиальный отрезок времени. Так же, экспоненциальные алгоритмы с возвратом могут занимать большое время.

1. **Какова должна быть размерность труднорешаемой задачи, чтобы имело смысл решать её с помощью алгоритма с возвратом?**

N < 20, при большем количестве, нужно разработать эвристические или приближенные алгоритмы

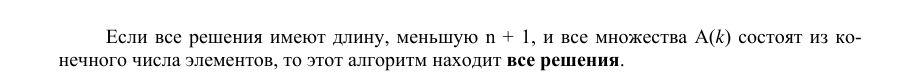
1. **За какое время алгоритм с возвратом может решить труднорешаемую задачу?**

При большой размерности, он может решать задачу за несколько веков, т.е. мы не сможем узнать ответ (если ли он или задача не решаема)

1. **Можно ли написать алгоритм с возвратом без рекурсии? Чем удобен рекурсивный вариант?**

Можно записать в виде не рекурсивного варианта, но удобство рекурсии в том, что возврат является частью механизма рекурсии.

1. **Сколько решений задачи можно найти с помощью алгоритма с возвратом?**

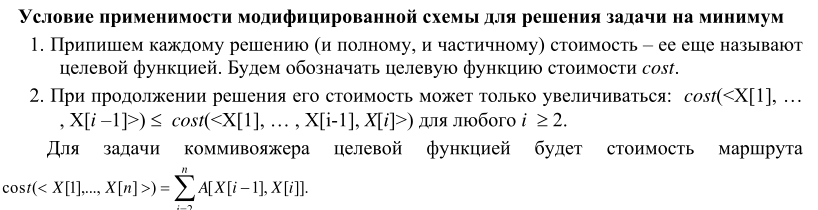


6. **Если *NP*-полная задача распознавания имеет ответ «нет», можно ли её решать алгоритмом с возвратом? Какое время потребуется для решения?**

Можно, но это будет не эффективно, алгоритм будет перебирать все варианты решения, и если размерность большая, то вычисление может занять очень больше время.

**7. В чём заключается модификация алгоритма с возвратом для решения задач на минимум?**

Модификация – алгоритм ветвей и границ.



**8. Какому условию должна удовлетворять труднорешаемая оптимизационная задача на минимум, чтобы её можно было решать модифицированной схемой алгоритма с возвратом?**

Размерность не должна превышать 20, так как при большем количестве алгоритм может выполняться очень долго.

**9. Можно ли решить задачу «КОММИВОЯЖЕР» с помощью общей (немодифицированной) схемы алгоритма с возвратом?**

Можно, но это будет не эффективно и решение будет найдено позже, нежели при использовании модифицированной схемы.

**10. Можно ли решить задачу «КОММИВОЯЖЕР» с помощью модифицированной схемы алгоритма с возвратом, если в графе есть отрицательные веса рёбер? Почему?**

Нельзя, т.к по условию задачи все расстояния положительные.

**11. Сколько контейнеров будет заполнено в худшем случае?**

Количество будет равно размерности (каждый предмет в отдельном контейнере)

**12. При достижении какого условия рекурсия остановится?**

Когда будут пройдены все ветви

**13. Перечислите последовательность шагов при реализации «прямого хода» решения.**

Проверяем текущий элемент на условие допустимости (вес текущего элемента меньше 1, а так же, если контейнер новый, то проверяем, меньше ли новая коробка + 1 чем текущее оптимальное количество коробок и предыдущий контейнер не пуст.

Пытаемся упаковать предмет в контейнер, если места не хватает, берем новый контейнер. Если найдено оптимальное решение, то сохраняем его, после чего осуществляем возврат на шаг назад.

**14. Каким образом и когда начинается перебор возможных вариантов упаковки?**

После того, как было соблюдено условие допустимости. После чего упаковываем предмет и если не выполнено условие (I < N) тогда текущее оптимальное количество коробок = текущему количеству коробок. После чего, осуществляем возврат на шаг назад.

**15. Какова сложность данного алгоритма?**

Экспоненциальная сложность.

1. **Назовите естественное ограничение данной задачи.**

Количество элементов, когда все элементы будут упакованы, то рекурсия закончится.

1. **Какое условие необходимо проверять на каждой итерации решения задачи?**

Проверяем текущий элемент на условие допустимости (вес текущего элемента меньше 1, а так же, если контейнер новый, то проверяем, меньше ли новая коробка + 1 чем текущее оптимальное количество коробок и предыдущий контейнер не пуст.

**8. В каком случае необходимо увеличить число использованных контейнеров?**

Если у нас не хватает места под элемент и новая коробка + 1 меньше чем текущее оптимальное кол-во коробок.