|  |  |
| --- | --- |
| 1.задача о загрузке рюкзака | динамическое программирование |
| 2.На палубе судна имеется мест для размещения стандартных контейнеров. Выбрать n контейнеров для погрузки на судно можно из имеющихся в наличии. Каждый контейнер характеризуется весом и доходом   от его перевозки. Необходимо выбрать контейнеров таким образом, чтобы их общий вес не превышал но при этом доход от перевозки был максимально возможным. Для какой задачи сформулировано это условие | О загрузке судна |
| 3.Оценочные ограничения строки i разрешающего столбца s  для симплекс - таблицы задача линейного программирования в следующие правила. | ¥, если bi =0 и ais<0  0, если bi =0 и ais>0 |
| 4.При решении задачи линейного программирования графическим методом оптимальным решением может быть. | одна точка  две точки  отрезок |
| 5.несбалансированная транспортная задача это | Открытая транспортная задача |
| 6.Если в транспортной задаче объем запасов превышает объем потребностей, в рассмотрение вводят | фиктивный пункт потребления |
| 7.Модель задачи линейного программирования, в которой целевая функция исследуется на максимум и система ограничений задачи является системой уравнений называется | канонической |
| 8.Модель задачи линейного программирования, в которой целевая функция исследуется на максимум и система ограничений задачи является системой неравенства называется | общей |
| 9.Общая задача линейного программирования может включать в себя: | систему ограничений в виде неравенств  систему ограничений в виде равенств  требования оптимизации линейной целевой функции |
| 10.Сетевой график имеет вид …. Время выполнения всего проекта равно | 17 |
| 11.Критический путь сетевого графика | это полный путь с макс продолжительностью |
| 12.полный путь сетевого графика – | это последовательность работ и событий, начинающаяся от исходного события и заканчивающаяся завершающим событием |
| 13.путь в сетевом графике | это любая непрерывная последовательность работ и событий |
| 14. Модель – это | аналог (образ) оригинала, но построенный средствами и методами отличными от оригинала + |
| 15.Сетевой график имеете вид … укажите пример полного пути | 1-3-6 |
| 16.коммивояжер должен посетить только один раз каждый из n городов и вернуться в исходный пункт. Его маршрут должен минимизировать суммарную длину пройденного пути это | задача коммивояжера |
| 17.Динамическое программирование | это метод оптимизации многошаговых задач в условиях мультипликативности и аддитивности целевой функции |
| 18.Распределенный метод решения транспортной задачи | поставка, передаваемая по циклу определяется как минимум среди поставок в клетках цикла со знаком "-" |
| 19.Метод потенциалов это | один из методов проверки опорного плана транспортной задачи на оптимальность |
| 20.Базисным решением системы m линейных уравнений с n переменными называется решение, в котором. | все n-m неосновных переменных равны нулю |
| 21.Транспортная задача является задачей …. Программирования | линейного |
| 22. В задачах линейного программирования, решаемых симплекс-методом искомые переменные должны быть | неотрицательными |
| 23.Алгоритм Джонсона Троттера применяется для | получения множества всех перестановок |
| 24.Будет большое условие задачи о рюкзаке, с весом, с вместимостью и т.д. | рюкзаке |
| 25.Решение задачи о загрузке судна основано на: | генераторе множества всех сочетаний; |
| 26.Решение задачи о коммивояжере основано на: | генераторе множества всех перестановок; |
| 27.метод ветвей и границ | это метод линейного программирования полного перебора |
| 28.В основе метода ветвей и границ лежат две процедуры: | ветвления и нахождения границ |
| 29.Решение упрощенной задачи о рюкзаке основано на: | генераторе множества всех подмножеств; |
| 30.Какую задачу нецелесообразно решать при помощи рекурсивных алгоритмов: | Линейного программирования |
| 31.дистанция левенштейна между словами сор и спорт равна | 2 |
| 32.Алгоритм поиска в ширину заключается в: | посещении вершин в порядке их удаленности от некоторой заранее выбранной или указанной стартовой вершины; |
| 33.К оптимизационным алгоритмам на графах не относятся: | Алгоритм поиска в высоту |
| 34.Теорема Форда-Фалкерсона: | В любой сети максимальная величина потока из истока s в сток t равна минимальной пропускной способности разреза отделяющего s от t; |
| 35.Сетевой график представляет собой | взвешенный ориентированный корневой граф без контуров (ациклический) и изолированных вершин, который построен по определенным правилам. |
| 36.Одним из возможных определений понятия линейное программирование является: | область математического программирования, посвященная теории и методам решения экстремальных задач, характеризующихся линейной зависимостью между переменными; |
| 37.При решении задачи линейного программирования геометрическим методом оптимальным решением может быть | Одна точка |
| 38.Задача, характеризующаяся тем, что целевая функция является линейной функцией переменных, а область допустимых значений определяется системой линейных равенств или неравенств, называется | Задача линейного программирования |
| 39.Вычислительный метод решения экстремальных задач определенной структуры, представляющий собой направленный последовательный перебор вариантов, который обязательно приводит к глобальному максимуму это | дискретное программирование |
| 40.Экономико-математическая модель | математическое представление экономической системы (объектов, задачи, явлений, процессов и т. п.) + |
| 41.Метод – это | подходы, пути и способы постановки и решения той или иной задачи в различных областях человеческой деятельности + |
| 42.Выберите неверное утверждение | ЭММ позволяют управлять объектом |
| 43.Найти экстремум функции f(x) при выполнении ограничений Ri(x) = ai, φ (x) ≤ bj, наложенных на параметры функции – это задача | условной оптимизации |
| 44.Задача, включающая целевую функцию f и функции Ф, входящие в ограничения, является задачей линейного программирования, если | все Ф и f являются линейными функциями относительно своих аргументов + |
| 45.Множество всех допустимых решений системы задачи линейного программирования | выпуклым |
| 46.Если задача линейного программирования имеет оптимальное решение, то целевая функция достигает нужного экстремального значения в одной из: | вершин многоугольника (многогранника) допустимых решений |
| 47.Симплексный метод решения задач линейного программирования включает: | определение одного из допустимых базисных решений поставленной задачи (опорного плана), определение правила перехода к не худшему решению, проверка оптимальности найденного решения |
| 48. Задача линейного программирования не имеет конечного оптимума, если | целевая функция не ограничена сверху на множестве допустимых решений |
| 49.При приведении задачи линейного программирования (ЛП) к виду основной задачи ЛП ограничения вида «< или =» преобразуются в ограничения равенства добавлением к его левой части дополнительной неотрицательной переменной. Вводимые дополнительные неизвестные имеют вполне определенный смысл. Так, если в ограничениях исходной задачи ЛП отражается расход и наличие производственных ресурсов, то числовое значение дополнительной переменной в решении задачи, записанной в виде основной имеет смысл | остатка ресурса |
| 50.Если ресурс образует «узкое место производства», то это означает | ресурс использован полностью |
| 51.Критерием остановки вычислений в алгоритме поиска оптимального решения методами одномерной оптимизации является условие | значение ЦФ, вычисленное в текущей точке, меньше значения ЦФ, вычисленного в предыдущей точке |
| 52.Если целевая функция и все ограничения выражаются с помощью линейных уравнений, то рассматриваемая задача является задачей | линейного программирования |
| 53.Задача линейного программирования может достигать максимального значения | во множестве точек |
| 54.В линейных оптимизационных моделях, решаемых с помощью геометрических построений число переменных должно быть | не больше двух |
| 55.Если в прямой задаче, какое-либо ограничение является неравенством, то в двойственной задаче соответствующая переменная | Неотрицательна |
| 56.Если в транспортной задаче объем спроса равен объему предложения, то такая задача называется | закрытой |
| 57.Методы теории игр предназначены для решения задач | с конфликтными ситуациями в условиях неопределенности |
| 58.Стратегия игрока – это совокупность правил, определяющих выбор его действий при | каждом ходе в зависимости от сложившейся ситуации в одном сеансе игры |
| 59.Нижняя цена игры – это | максимин, т.е. максимальный выигрыш по всем стратегиям одного из игроков среди минимальных значений выигрышей каждой его стратегии |
| 60.Верхняя цена игры – это | минимакс, т.е. минимальный проигрыш по всем стратегиям одного из игроков среди максимальных значений проигрышей каждой его стратегии |
| 61.Решение игры в чистых стратегиях определяется | вероятностью выбора каждой из активных (полезных) стратегий, совокупный выигрыш которых представляет случайную величину с математическим ожиданием равным цене игры + |
| 62.Задача, процесс нахождения решения которой является многоэтапным, относится к задачам | динамического программирования |
| 63.Выберите правильный вариант. Задача линейного программирования не имеет конечного оптимума, если: | целевая функция не ограничена сверху на множестве допустимых решений |
| 64.Особенности модели динамического моделирования: | задача оптимизации интерпретируется как многошаговый процесс управления; целевая функция равна сумме целевых функций каждого шага |
| 65.Общая задача целочисленного программирования: Найти такое решение X=(x1,...,xn), при котором линейная функция Z=Scjxj принимает минимальное или максимальное значение при ограничениях: | xj  0, xj - целые |
| 66.Для Марковского процесса в физической системе характерно: | для каждого момента времени вероятность любого состояния системы в будущем зависит только от состояния системы в настоящий момент |
| 67.Задачи теории массового обслуживания: | определение необходимой скорости обслуживания  рациональное построение очереди |
| 68.В задаче многокритериальной оптимизации для оценки качества найденных решений используют эталонные точки: | утопическая точка  надир |
| 69.Методы отсечения: | сначала задача решается без условия целочисленности вводится дополнительное ограничение правильности отсечения |
| 70.Транспортная задача. Найти объемы перевозок для каждой пары "поставщик" - "потребитель" так, чтобы: | мощности всех поставщиков были реализованы  спросы всех потребителей были удовлетворены  суммарные затраты на перевозку были минимальны |
| 71.Математическая постановка задачи оптимального уравнения включает следующие элементы | математическое описание объекта управления  описание управляющего воздействия  математическое описание критерия качества управления  описание изменения (движения) объекта управления |
| 72.Пусть решается задача определенного экстремума. Составим функцию Лагранжа:  **L(x1,...,xn)=f(x1,...,xn)+ii(x1,...,xn)**. Для определения стационарных точек необходимо. | приравнять к нулю производные L по переменным x1,...,xn |
| 73.Задачи конечномерной оптимизации делятся на ... | Точные  эвристические |
| 74.Согласно первой теореме двойственности: | если одна задача имеет оптимальное решение, то двойственная задача тоже имеет оптимальное решение  если линейная функция одной из задач не ограничена, то условия двойственной задачи противоречивы |
| 75.Метод северо-западного угла: "поставщик" - "потребитель" так, чтобы: | переменной x11 дается максимально возможное значение  после вычеркивания первого столбца северо-западным элементом будет является элемент x12 |
| 76.Для взаимно-двойственных задач линейного программирования. | в одной задаче ищется максимум в другой – минимум  матрицы коэффициентов при переменных в системах ограничений обеих задач являются транспонированными друг другу |
| 77.Критерий оптимальности решения задачи линейного программирования при отыскании максимума линейной функции с выражением линейной функции через неосновные переменные ..., то решение задачи оптимально | отсутствуют положительные коэффициенты при неосновных переменных |
| На рисунке изображена схема, поясняющая решение о загрузке судна с условием центровки. Задача имеет следующие исходные данные: n = 4 – общее количество контейнеров; m = 3 – количество свободных мест на палубе судна; (100,200,300,400) – вес контейнеров; (10,15,20,25) – доход от перевозки контейнеров; (350,250,0) – минимальный вес контейнеров;(750,350,750) – максимальный вес контейнеров. В какой строке представлен оптимальный план размещения контейнеров на палубе? | 22  (контейнеры 3 2 1, доход от перевозки 60) |
| Комбинаторный анализ – | это раздел математики, посвященный решению задач выбора и расположения элементов некоторого, множества в соответствии с заданными правилами. |
| Из скольких этапов состоит решение транспортной задачи: | 2 |
| Алгоритм поиска в ширину заключается в: | **посещении вершин в порядке их удаленности от некоторой заранее выбранной или указанной стартовой вершины;** |
| Какую задачу нецелесообразно решать при помощи динамического программирования? | **Линейного программирования** |
| Имеется множество, состоящее из n элементов. Количество элементов множества всех подмножеств вычисляется по формуле | **2^n** |
| Имеется множество, состоящее из n элементов. Количество перестановок данного множества рассчитывается по формуле? | **n!** |
| Математическое программирование – это? | **область математики, изучающая теорию и численные методы решения многомерных экстремальных задач** |

|  |
| --- |
|  |
|  |
|  |

1. Ниже представлен фрагмент программного кода, реализующую функцию knapsack\_s. Для решения какой задачи используется эта функция? **упрощенной задачи о рюкзаке;**

// Knapsack.cpp  
#include "stdafx.h"  
#include "Knapsack.h"  
#define NINF 0x80000000   
int calcv(combi::subset s, const int v[])   
{  
int rc = 0;   
for (int i = 0; i < s.sn; i++) rc += v[s.ntx(i)];   
return rc;  
};  
int calcc(combi::subset s, const int v[], const int c[])   
{  
int rc = 0;   
for (int i = 0; i < s.sn; i++) rc += (v[s.ntx(i)]\*c[s.ntx(i)]);   
return rc;  
};  
void setm(combi::subset s, short m[])  
{  
for (int i = 0; i <s.n; i++) m[i] = 0;  
for (int i = 0; i < s.sn; i++) m[s.ntx(i)] = 1;  
};  
int knapsack\_s(  
int V,   
short n,   
const int v[],   
const int c[],   
short m[]   
)  
{  
combi::subset s(n);  
int maxc = NINF, cc = 0;  
short ns = s.getfirst();   
while (ns >= 0)  
{  
if (calcv(s, v) <= V)  
if ((cc = calcc(s,v,c)) > maxc)   
{  
maxc = cc;  
setm(s,m);  
}  
ns = s.getnext();   
};  
return maxc;   
};

1. Фрагмент программного кода, представленный ниже, используется для программной реализации?

**генератора множества всех сочетаний**

// Combi.cpp   
#include "stdafx.h"  
#include "Combi.h"  
#include <algorithm>  
namespace combi  
{xcombination::xcombination (short n, short m)  
{  
this->n = n;  
this->m = m;  
this->sset = new short[m+2];  
this->reset();   
}void xcombination::reset() // сброситьгенератор, начатьсначала  
{this->nc = 0;  
for(int i = 0; i < this->m; i++) this->sset[i] = i;   
this->sset[m] = this->n;  
this->sset[m+1] = 0;   
}; short xcombination::getfirst()  
{ return (this->n >= this->m)?this->m:-1; };  
short xcombination::getnext() // сформировать следующий массив индексов  
{short rc = getfirst();   
if (rc > 0)  
{short j;  
for (j = 0; this->sset[j]+1 == this->sset[j+1]; ++j)   
this->sset[j] = j;  
if (j >= this->m) rc = -1;  
else {  
this->sset[j]++;  
this->nc++;  
};}return rc;   
}; short xcombination::ntx(short i)  
{return this->sset[i]; };  
unsigned \_\_int64 fact(unsigned \_\_int64 x){return(x == 0)?1:(x\*fact(x-1));};  
unsigned \_\_int64 xcombination::count() const  
{return (this->n >= this->m)?  
fact(this->n)/(fact(this->n-this->m)\*fact(this->m)):0;   
};

1. Ниже представлен фрагмент программного кода функции boat\_c, применяемой при решении задачи о загрузке судна с условием центровки. Укажите какое из утверждений является неправильным:

**В функции boat\_c применяется генератор размещений (combi::accomodation). Конструктору генератора передается один параметр: n(количество контейнеров). Функция boat\_c вызывает вспомогательную функцию boatfnc::compv, которая позволяет проверить допустимость размещения контейнеров. Если вес всех контейнеров в размещении удовлетворяет заданным ограничениям, то функция возвращает true, иначе возвращается false.**

// --- Вoat.cpp  
#include "stdafx.h"  
#include "Boat.h"  
#include "Combi.h"  
namespace boatfnc  
{bool compv( combi::accomodation s, const int ming[],   
const int maxg[], const int v[])  
{int i = 0;  
while(i < s.m && v[s.ntx(i)] <= maxg[i] && v[s.ntx(i)] >= ming[i])i++;   
return (i == s.m);};  
int calcc(combi::accomodation s,const int c[])   
{int rc = 0;   
for (int i = 0; i < s.m; i++) rc += c[s.ntx(i)];   
return rc;};  
voidcopycomb(short m, short \*r1, const short \*r2)  
{ for (int i = 0; i < m; i++) r1[i] = r2[i]; };}  
intboat\_с( // функция возвращает доход от перевозки контейнеров  
short m,// [in] количество мест для контейнеров  
intminv[], // [in] минимальный вес контейнера на каждом месте  
intmaxv[], // [in] максимальный вес коннтейнера каждом месте  
short n,// [in] всего контейнеров  
const int v[],// [in] вес каждого контейнера  
const int c[],// [in] доход от перевозки каждого контейнера  
short r[]// [out] номера выбранных контейнеров)  
{combi::accomodation s(n, m);  
int rc = 0, i = s.getfirst(), cc = 0;   
while (i > 0)  
{if (boatfnc::compv(s, minv, maxv, v))   
if ((cc = boatfnc::calcc(s,c)) > rc)   
{rc = cc; boatfnc::copycomb(m, r, s.sset);}  
i = s.getnext(); };  
return rc;};

|  |  |
| --- | --- |
| 1. На рисунке представлено изображение некого графа     Данный граф является: | **ориентированным;** |
| 1. Ниже представлен фрагмент программного кода, обеспечивающий вычисление продолжительности решения задачи о рюкзаке при различном количестве предметов. Какая стандартная функция отвечает за вычисление продолжительности решения задачи? | **clock();** |
| 1. Имеется некая задача линейного программирования с двумя неизвестными. Какими методами следует решать данную задачу: | **графическим и симплекс-методом**; |
| 1. Будет нарисован граф. Попросят указать полный путь | **Если задания такие же, то это 1-3-6.**  **А делай так – ищи исходную, конечную и промежуточные**  **вершины** |
| 1. Будет нарисован граф.   Попросят указать  критический путь (суммарный вес) | **Если задания такие же, то это 40.**  **Пр просто выбирай полный путь с**  **самой максимальной суммой**  **и пропускных способностей** |
| 1. К какому типу относится ТЗ (транспортная задача) | **к линейному программированию** |
| 1. в транспортной задаче переизбыток продавцов | **+** |
| 1. Симлекс-метод | **Симплекс-метод — алгоритм решения** [**оптимизационной**](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) **задачи** [**линейного программирования**](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) **путём перебора вершин выпуклого многогранника в многомерном пространстве. Сущность метода: построение базисных решений, на которых монотонно убывает линейный функционал, до ситуации, когда выполняются необходимые условия локальной оптимальности.** |
| 1. В общем виде модель задачи математического программирования выглядит следующим образом (где х – искомая, в общем случае векторная, величина; Х – область определения искомой величины; функция цели (функция определяющая значение критерия оптимальности)): | **Z(x) → (min,max)** |
| 1. permutation::permutation(short n) | **перестановок** |
| 1. Имеется множество, состоящее из 3-х элементов {A, B, C}.Множество всех сочетаний по 2 элемента данного множества имеет вид | **{B,C}, {A,C},{A,B}** |
| 1. Одно из определений рекурсивного алгоритма звучит как: | **алгоритм, решающий задачу путем сведения ее к решению одной или нескольких таких же задач, но в сокращенном их варианте;** |
| 1. Имеется множество, состоящее из 3-х элементов {A, B, C}. Множество всех подмножеств данного множества | **{}, {C}, {B}, {B,C}, {A}, {A,C}, {A,B}, {A,B,C}** |
| 1. Имеется множество, состоящее из n элементов. Количество сочетаний элементов мощностью m вычисляется по формуле | **n!/((n-m)!m!)** |
| 1. При решении задачи вычисления дистанции Левенштейна с одинаковыми исходными данными при помощи динамического программирования и рекурсивных алгоритмов какой из методов решения является более быстрым: | **динамического программирования;** |
| 1. Ниже представлен фрагмент программного кода функции boat\_c, применяемой при решении задачи о загрузке судна с условием центровки. Укажите какое из утверждений является неправильным: | **В функции boat применяется генератор размещений (combi::accommodation). Конструктору генератора передается один параметр: n(количество контейнеров).** |
| 1. Алгоритм поиска в глубину заключается в: | **том, чтобы идти вперед в неисследованную область, пока это возможно, если же вокруг все исследовано, отступить на шаг назад и искать новые возможности для продвижения вперед;** |
| 1. Одним из возможных определений понятия нелинейное программирование является: | **область математического программирования, посвященная теории и методам решения экстремальных задач, характеризующихся нелинейной зависимостью между переменными** |
| 1. Назначение транспортной задачи: | **Определение объемов перевозок из пунктов отправления в пункты назначения с минимальной суммарной стоимостью перевозок** |
| 1. Что из перечисленного не относится к задачам математического программирования | **C++ программирование** |
| 1. К несмежным дисциплинам математического программирования относится | **web-дизайн** |
| 1. Решение задачи математического программирования осуществляется в 4 этапа | **1. Построение математической модели**  **2. Классификация задачи**  **3. Выбор метода решения**  **4. Вычисление** |
| 1. Имеется множество, состоящее из n элементов. Множество размещений множества n по m элементов рассчитывается по формуле | **n!/(n-m)!** |
| 1. Имеется множество, состоящее из 3-х элементов {A, B, C}. Множество всех перестановок имеет вид | **{abc}, {acb}, { bac}, {bca}, { cab}, {cba}.** |
| 1. Имеется множество, состоящее из 3-х элементов {A, B, C}. Множество всех размещений по 2 элемента имеет вид | **{B,C}, {C,B}, {A,C},{C,A}; {A,B}, {B,A}** |
| 1. Алгоритм топологической сортировки заключается в: | **упорядочивания вершин бесконтурного ориентированного графа согласно линейного порядка** |
| 1. На рисунке представлена фрагмент программы, позволяющей оценить продолжительность решения задачи коммивояжера в зависимости от количества городов и результат выполнения программы. В каких единицах отображена продолжительность вычисления? | **условные единицы процессорного времени;** |
| 1. Задача о загрузке судна имеет следующие исходные данные: V = 1000 – ограничение по общему весу контейнеров; n = 6 – количество контейнеров; m = 3 – количество свободных мест на палубе; (100,200,300,400,500,150) – вес контейнеров; (10,15,20,25,30,25) – доход от перевозки контейнеров. На рисунке представлена схема решения. Какая строка на данном рисунке соответствует решению задачи? | **18** |
| 1. Граф решения какой задачи представлен рисунке: | **о коммивояжере** |
| 1. Даны исходные данные для решения транспортной задачи: m– количество поставщиков продукции; n– количество потребителей продукции; i– индекс для поставщиков; j– индекс для потребителей; ai – наличие продукции у каждого поставщика; bj – потребность в продукции каждого потребителя; cij– стоимость доставки продукции единицы продукции от I поставщика к j потребителю. Как будет выглядеть целевая функция при построении математической модели задачи: | **Z(x) =** |
| 1. Задача о коммивояжере формулируется следующим образом: коммивояжер (бродячий торговец) должен найти минимальный кольцевой маршрут обхода n городов. Расстояние d между каждой парой городов считается известным. Как может быть записана математическая модель задачи (ki – неизвестные (номера выбранных городов), которые требуется найти)? |  |
| 1. Условие задачи о загрузке судна выглядит следующим образом:  На палубе судна имеется m мест для размещения стандартных контейнеров. Выбрать n контейнеров для погрузки на судно можно из n больше m имеющихся в наличии. Каждый контейнер i характеризуется весом v и доходом c от его перевозки. Необходимо выбрать m контейнеров таким образом, чтобы их общий вес не превышал V, но при этом доход от перевозки был максимально возможным. Как может быть записана математическая модель задачи (ki – неизвестные (номера выбранных контейнеров), которые требуется найти): | **Ответ в картинке** |
| 1. функция salesman (верное утверждение) | **Функция salesman возвращает доход коммивояжера или значение INF, что обозначает отсутствие кольцевых маршрутов.** |
| 1. Графическая реализация. (При решении задачи линейного программирования графическим методом оптимальным решением может быть)? | **Все правильно, КРОМЕ «интервалы». (1 точка, 2 точки, отрезок)** |
| 1. Модель задачи линейного программирования, в которой целевая функция исследуется на максимум и система ограничений задачи является системой ***уравнений*** называется? | **Каноническая** |
| 1. Посчитать полный путь для сетевого графа. (Он нарисован будет) | **17** |
| 1. Вопрос про цикл обхода (ставятся плюс и минус)? | **поставка, передаваемая по циклу, определяется как минимум среди поставок в клетках цикла со знаком "-"** |
| 1. Переменные в линейной зависимости | **Не отрицательные** |
| 1. В линейной задаче какие условия накладываются (вроде тоже не было) | **- переменные, которые следует определить;**  **- целевую функцию, подлежащую оптимизации;**  **- систему ограничений в форме линейных уравнений и неравенств.** |