|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Математическое программирование - это | RadioButton | область математики, изучающая теорию и численные методы решения многомерных экстремальных задач раздел кибернетики, изучающий теорию чисел раздел программирования, изучающий способы написания программного кода для решения математических задач область общей теории программирования, изучающая способы написания программного кода с помощью математических функций | 1 | 2 |
|  | Решение задачи математического программирования осуществляется в 4 этапа | RadioButton | выбор метода решения, построение математической модели, классификация задачи, вычисление построение математической модели, классификация задачи, выбор метода решения, вычисление построение математической модели, написание программного кода, вычисление написание программного кода, построение математической модели, вычисление | **ОТВЕТЫ**  2 | 2 |
|  | Что из перечисленного не относится к задачам математического программирования | RadioButton | целочисленное программирование линейное программирование С++ программирование дискретное программирование | 3 | 2 |
|  | К несмежным дисциплинам математического программирования относится | RadioButton | математическое моделирование web-дизайн методы оптимизации исследование операций | 2 | 1 |
|  | Комбинаторный анализ - это | RadioButton | раздел математики, посвященный решению задач выбора и расположения элементов некоторого, множества в соответствии с заданными правилами раздел информатики, изучающий правильность написания программных кодов для различных комбинаций алгоритмов раздел математики, посвященный решению задач нелинейного программирования раздел информатики, изучающий правильность подключения комбинаций компьютеров в локальной сети | 1 | 3 |
|  | Имеется множество, состоящее из n элементов. Количество элементов множества всех подмножеств вычисляется по формуле | RadioButton | n! 2^n n!/((n-m)!m!) n!/(n-m)! | 2 | 4 |
|  | Имеется множество, состоящее из n элементов. Количество сочетаний элементов мощностью m вычисляется по формуле | RadioButton | 2^n n! n!/(n-m)! n!/((n-m)!m!) | 4 | 4 |
|  | Имеется множество, состоящее из nэлементов. Количество перестановок данного множества рассчитывается по формуле | RadioButton | n!/((n-m)!m!) 2^n n!/(n-m)! n! | 4 | 4 |
|  | Имеется множество, состоящее из n элементов. Множество размещений множества n по m элементов рассчитывается по формуле | RadioButton | n!/(n-m)! n!/((n-m)!m!) 2^n n! | 1 | 4 |
|  | Имеется множество, состоящее из 3-х элементов {A, B, C}. Множество всех подмножеств данного множества | RadioButton | {B,C}, {C,B}, {A,C},{C,A}; {A,B}, {B,A} {B,C}, {A,C},{A,B} {A,B,C}, {A,C,B}, {C,A,B}, {C,B,A},{B,C,A},{B,A,C} {}, {C}, {B}, {B,C}, {A}, {A,C}, {A,B}, {A,B,C} | 4 | 4 |
|  | Имеется множество, состоящее из 3-х элементов {A, B, C}.Множество всех сочетаний по 2 элемента данного множества имеет вид | RadioButton | {B,C}, {C,B}, {A,C}, {C,A}; {A,B}, {B,A} {}, {C}, {B}, {B,C}, {A}, {A,C}, {A,B}, {A,B,C} {B,C}, {A,C},{A,B} {A,B,C}, {A,C,B}, {C,A,B}, {C,B,A}, {B,C,A},{B,A,C} | 3 | 4 |
|  | Имеется множество, состоящее из 3-х элементов {A, B, C}. Множество всех перестановок имеет вид | RadioButton | {B,C}, {A,C},{A,B} {A,B,C}, {A,C,B}, {C,A,B}, {C,B,A},{B,C,A},{B,A,C} {}, {C}, {B}, {B,C}, {A}, {A,C}, {A,B}, {A,B,C} {B,C}, {C,B}, {A,C},{C,A}; {A,B}, {B,A} | 2 | 4 |
|  | Имеется множество, состоящее из 3-х элементов {A, B, C}. Множество всех размещений по 2 элемента имеет вид | RadioButton | {B,C}, {C,B}, {A,C},{C,A}; {A,B}, {B,A} {}, {C}, {B}, {B,C}, {A}, {A,C}, {A,B}, {A,B,C} {B,C}, {A,C},{A,B} {A,B,C}, {A,C,B}, {C,A,B}, {C,B,A},{B,C,A},{B,A,C} | 1 | 4 |
|  | Алгоритм Джонсона-Троттера применяется для получения | RadioButton | Множества всех подмножеств Множества всех сочетаний Множества всех перестановок Множества всех размещений | 3 | 2 |
|  | Фрагмент программного кода, представленный ниже используется для программной реализации // Combi.cpp  #include "stdafx.h" #include "Combi.h" #include <algorithm> namespace combi { subset::subset(short n)  { this->n = n; this->sset = new short[n];  this->reset();  }; void subset::reset()  { this->sn = 0; this->mask = 0;  };  short subset::getfirst()  { \_\_int64 buf = this->mask;  this->sn = 0; for (short i = 0; i < n; i++) { if (buf & 0x1) this->sset[this->sn++] = i;   buf >>= 1;  } return this->sn;  }; short subset::getnext()  { int rc = - 1; this->sn = 0; if (++this->mask < this->count()) rc = getfirst(); return rc;  }; short subset::ntx(short i) {return this->sset[i];};  unsigned \_\_int64 subset::count() {return (unsigned \_\_int64)(1<<this->n);};  }; | RadioButton | генератора множества всех размещений генератора множества всехподмножеств генератора множества всех сочетаний генератора множества всех перестановок | 2 | 5 |
|  | Фрагмент программного кода, представленный ниже, используется для программной реализации // 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;  }; }; | RadioButton | генератора множества всех перестановок генератора множества всех размещений генератора множества всех подмножеств генератора множества всех сочетаний | 4 | 5 |
|  | Фрагмент программного кода, представленный ниже, используется для программной реализации // Combi.cpp  #include "stdafx.h" #include "Combi.h" #include <algorithm> #define NINF ((short)0x8000) namespace combi { permutation::permutation(short n)  { this->n = n; this->sset = new short[n]; this->dart = new bool[n]; this->reset();  };  voidpermutation::reset()  { this->getfirst(); };  \_\_int64 permutation::getfirst()  { this->np = 0;  for (int i = 0; i < this->n; i++) {this->sset[i] = i; this->dart[i] = L;}; return (this->n > 0)?this->np:-1; }; \_\_int64 permutation::getnext() //  { \_\_int64 rc = - 1; short maxm = NINF, idx = -1;  for(int i = 0; i < this->n; i++) { if ( i > 0 && this->dart[i] == L && this->sset[i] > this->sset[i-1] && maxm < this->sset[i]) maxm = this->sset[idx = i]; if ( i < (this->n-1)&& this->dart[i] == R && this->sset[i] > this->sset[i+1]&& maxm < this->sset[i]) maxm = this->sset[idx = i];  }; if (idx >= 0) {  std::swap(this->sset[idx], this->sset[idx+(this->dart[idx]== L?-1:1)]);  std::swap(this->dart[idx], this->dart[idx+(this->dart[idx]== L?-1:1)]);  for (int i = 0; i < this->n; i++)  if (this->sset[i] > maxm) this->dart[i] = !this->dart[i]; rc = ++this->np; }  return rc; }; short permutation::ntx(short i){return this->sset[i];};  unsigned \_\_int64 fact(unsigned \_\_int64 x){return (x == 0)?1:(x\*fact(x-1));}; unsigned \_\_int64 permutation::count() const {return fact(this->n); }; } | RadioButton | генератора множества всех перестановок генератора множества всех размещений генератора множества всех подмножеств генератора множества всех сочетаний | 1 | 5 |
|  | Фрагмент программного кода, представленный ниже, используется для программной реализации // Combi.cpp #include "stdafx.h" #include "Combi.h" namespace combi { accomodation::accomodation (short n, short m) { this->n = n; this->m = m; this->cgen = new xcombination(n,m); this->pgen = new permutation(m);  this->sset = new short[m]; this->reset();  } void accomodation::reset() { this->na = 0; this->cgen->reset();  this->pgen->reset();  this->cgen->getfirst(); };  short accomodation::getfirst() { short rc = (this->n >= this->m)?this->m:-1; if (rc > 0)  { for (int i = 0; i <= this->m; i++)  this->sset[i] = this->cgen->sset[this->pgen->ntx(i)]; }; return rc;  }; short accomodation::getnext() { short rc; this->na++; if ((this->pgen->getnext())> 0)rc = this->getfirst(); else if ((rc = this->cgen->getnext())> 0)  {this->pgen->reset(); rc = this->getfirst();}; return rc;  }; short accomodation::ntx(short i) {return this->sset[i]; };  unsigned \_\_int64 fact(unsigned \_\_int64 x){ return (x == 0)?1:(x\*fact(x-1));};  unsigned \_\_int64 accomodation::count() const {  return (this->n >= this->m)? fact(this->n)/fact(this->n - this->m):0; }; | RadioButton | генератора множества всех перестановок генератора множества всех размещений генератора множества всех подмножеств генератора множества всех сочетаний | 2 | 5 |
|  | Ниже представлен фрагмент программного кода, реализующего алгоритм получения множества всех подмножеств и результат работы программы. Какие строки программного кода отвечает за появление на экране надписи, обозначенной рамкой синего цвета: 1// Main 2#include "stdafx.h" 3#include <iostream> 4#include "Combi.h" 5int \_tmain(int argc, \_TCHAR\* argv[]) 6{ 7 setlocale(LC\_ALL, "rus"); 8 char AA[][2]= {"A", "B", "C", "D"};  9std::cout<<std::endl<<" - Генератор множества всех подмножеств -"; 10 std::cout<<std::endl<<"Исходное множество: "; 11std::cout<<"{ "; 12 for (int i = 0; i < sizeof(AA)/2; i++)  13 std::cout<<AA[i]<<((i< sizeof(AA)/2-1)?", ":" ");  14std::cout<<"}"; 15 std::cout<<std::endl<<"Генерация всех подмножеств"; 16 combi::subset s1(sizeof(AA)/2);  17intn = s1.getfirst();  18while (n>= 0)  19 { 20std::cout<<std::endl<<"{ "; 21 for (int i = 0; i < n; i++)  22 std::cout<<AA[s1.ntx(i)]<<((i< n-1)?", ":" ");  23std::cout<<"}"; 24 n = s1.getnext();  25}; 26 std::cout<<std::endl<<"всего: " << s1.count()<<std::endl; 27system("pause"); 28 return 0; 29}  <http://portal.by/uploads/0d7a9b98254cb7c038d4310f2f3dd036.JPG> | RadioButton | 20-23 8,9 11-14 25-27 | 3 | 3 |
|  | Ниже представлен фрагмент программного кода, реализующую функцию knapsack\_s. Для решения какой задачи используется эта функция?  // 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;  }; | RadioButton | генерации множества всех подмножеств; о загрузке судна; упрощенной задачи о рюкзаке; о коммивояжере; | 3 | 4 |
|  | Решение упрощенной задачи о рюкзаке основано на: | RadioButton | генераторе множества всех подмножеств; генераторе множества всех размещений; генераторе множества всех перестановок; генераторе множества всех сочетаний; | 1 | 4 |
|  | 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? | RadioButton | system("pause"); maxcc; cout; clock(); | 4 | 3 |
|  | Условие задачи: 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. Для какой задачи сформулировано это условие? | RadioButton | упрощенной задачи о рюкзаке; генерации всех сочетаний; о загрузке судна; о загрузке судна с условием центровки; | 3 | 3 |
|  | Решение задачи о загрузке судна основано на: | RadioButton | генераторе множества всех перестановок; генераторе множества всех сочетаний; генераторе множества всех подмножеств; генераторе множества всех размещений; | 2 | 4 |
|  | Ниже представлена программная реализация функции boat, применяемой при решении задачи о загрузке судна. Укажите ошибочное утверждение: // --- Вoat.cpp #include "stdafx.h" #include "Boat.h" namespace boatfnc { int calcv(combi::xcombination s, const int v[]) // вес { int rc = 0;  for (int i = 0; i < s.m; i++) rc += v[s.ntx(i)];  return rc; };  int calcc(combi::xcombination s, const int c[]) // доход { int rc = 0;  for (int i = 0; i < s.m; i++) rc += c[s.ntx(i)];  return rc; };  void copycomb(short m, short \*r1, const short \*r2) // копировать { for (int i = 0; i < m; i++) r1[i] = r2[i];  };  } int boat( int V, // [in] максимальный вес груза short m, // [in] количество мест для контейнеров short n, // [in] всего контейнеров const int v[], // [in] вес каждого контейнера const int c[], // [in] доход от перевозки каждого контейнера short r[] // [out] результат: индексы выбранных контейнеров ) { combi::xcombination xc(n, m); int rc = 0, i = xc.getfirst(), cc = 0;  while (i > 0) { if (boatfnc::calcv(xc,v)<= V)  if ((cc = boatfnc::calcc(xc,c)) > rc)  {rc = cc; boatfnc::copycomb(m, r, xc.sset);} i = xc.getnext();  }; return rc; }; | RadioButton | В процессе своей работы функция boat вызывает вспомогательную функциюboatfnc::calcс (расчет дохода от транспортировки текущего сочетания контейнеров); В процессе своей работы функция boat вызывает вспомогательную функциюboatfnc::calcv (расчет веса текущего сочетания контейнеров); В процессе своей работы функция boat использует генератор размещений (combi::xcombination); В процессе своей работы функция boat вызывает вспомогательную функциюboatfnc::copycomb(копирование текущей комбинации); | 3 | 3 |
|  | Решение задачи о коммивояжере основано на: | RadioButton | генераторе множества всех перестановок; генераторе множества всех сочетаний; генераторе множества всех подмножеств; генераторе множества всех размещений; | 1 | 3 |
|  | На рисунке представлен фрагмент программного кода реализации функции salesman для решения задачи о коммивояжере. Какое из утверждений является неправильным: // -- Salesman.cpp #include "stdafx.h" #include "Salesman.h" int sum (int x1, int x2) // суммированиесучетомбесконечности { return (x1 == INF || x2 == INF)? INF: (x1 + x2); }; int\* firstpath(intn) // формирование 1гомаршрута 0,1,2,..., n-1, 0 {  int \*rc = new int[n+1];rc[n] = 0; for (int i = 0; i < n; i++) rc[i] = i;  returnrc; }; int\* source(intn) // формирование исходного массива 1,2,..., n-1 {  int \*rc = new int[n-1];  for (int i = 1; i < n; i++) rc[i-1] = i; return rc;  }; void copypath(int n, int \*r1, const int \*r2) // копироватьмаршрут { for (int i = 0; i < n; i++) r1[i] = r2[i]; }; int distance(int n, int \*r, const int \*d) // длинамаршрута { int rc = 0;  for (int i = 0; i < n-1; i++) rc = sum(rc, d[r[i]\*n+r[i+1]]);  returnsum (rc, d[r[n-1]\*n + 0]); //+последняядуга (n-1,0) }; void indx(int n, int \*r, const int \*s, const short \*ntx)  { for (int i = 1; i < n; i++) r[i] = s[ntx[i-1]];} int salesman (  intn, // [in] количествогородов constint \*d, // [in] массив [n\*n] расстояний int \*r // [out] массив [n] маршрут 0 x x x x ) { int \*s = source(n), \*b = firstpath(n), rc = INF, dist = 0;  combi::permutation p(n-1); int k = p.getfirst(); while (k >= 0)// цикл генерации перестановок {  indx(n, b, s, p.sset); // новый маршрут if ((dist = distance(n,b,d)) < rc) {rc = dist; copypath(n,r,b);} k = p.getnext();  }; return rc;  } | RadioButton | В функции salesmanприменяется генератор перестановок; Функция вызывает шесть вспомогательных функций: indx (формирование перестановки городов на основе массива идексов), distance(вычисление длины кольцевого маршрута), copypath(применение генератора перестановок), source (формирование исходного массива), firstpath(формирование первого маршрута) и sum (суммирование двух чисел с учетом того, что одно из них может быть равно бесконечности). Функция salesmanв цикле генерирует все возможные кольцевые маршруты, вычисляет для каждого маршрута длину (функция distance), фиксирует оптимальный маршрут (функция copypath). Функция salesman возвращает доход коммивояжера или значение INF, что обозначает отсутствие кольцевых маршрутов. | 4 | 5 |
|  | Ниже представлен фрагмент программного кода функции boat\_c, применяемой при решении задачи о загрузке судна с условием центровки. Укажите какое из утверждений является неправильным: // --- 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; }; | RadioButton | Функция boat\_c вызывает вспомогательную функцию boatfnc::calcc, которая дает возможность вычислить доход от перевозки контейнеров при заданном размещении. В функции boat\_cприменяется 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. Функция boat\_c вызывает вспомогательную функцию boatfnc::copycomb, которая предназначена для сохранения (копирования) наиболее доходного на данном шаге допустимого размещения. | 2 | 4 |
|  | Метод ветвей и границ – это: | RadioButton | общий алгоритмический метод решения задач нелинейного программирования, который по существу является вариацией полного перебора с отсевом подмножеств допустимых решений, заведомо не содержащих оптимальных решений. общий алгоритмический метод решения задач линейного программирования, который по существу является вариацией полного перебора с отсевом подмножеств допустимых решений, заведомо не содержащих оптимальных решений. общий алгоритмический метод решения задач комбинаторной оптимизации, который по существу является вариацией метода случайного поиска. общий алгоритмический метод решения задач комбинаторной оптимизации, который по существу является вариацией полного перебора с отсевом подмножеств допустимых решений, заведомо не содержащих оптимальных решений. | 4 | 3 |
|  | В основе метода ветвей и границ лежат две процедуры: | RadioButton | ветвления и разбиения на множества; ветвления и вычисления нижней или верхней границы; вычисления нижней или верхней границы и выбора оптимальной границы; ветвления и разбиения на подмножества; | 2 | 3 |
|  | Одно из определений рекурсивного алгоритма звучит как: | RadioButton | алгоритм, решающий задачу путем простого перебора всех возможных вариантов решений; алгоритм, решающий задачу путем выбора всех возможных вариантов решений случайным образом; алгоритм, решающий задачу путем сведения ее к решению одной или нескольких таких же задач, но в сокращенном их варианте; алгоритм, решающий задачу при помощи симплекс-метода; | 3 | 2 |
|  | Какую задачу нецелесообразно решать при помощи рекурсивных алгоритмов: | RadioButton | о рюкзаке; линейного программирования; о расстановке скобок при умножении матрицы; вычисления дистанции Левенштейна; | 2 | 3 |
|  | Дистанция Левенштейна между словами «сор» и «спорт» равна: | RadioButton | 2; 5; 4; 3; | 1 | 3 |
|  | Дистанция Левенштейна между словами «сорт» и «спорт» равна: | RadioButton | 1; 2; 3; 4; | 1 | 2 |
|  | Одним из определений динамического программирования является: | RadioButton | Метод решения задачи оптимизации, реализующей рекурсивный алгоритм с перекрывающимися подзадачами, в котором каждая такая подзадача решается один раз, а ее результат сохраняется для последующего применения. Метод решения задачи оптимизации, реализующей полный перебор, в котором каждая такая подзадача решается один раз, а ее результат сохраняется для последующего применения. Метод решения задачи линейного программирования, реализующей полный перебор, в котором каждая такая подзадача решается один раз, а ее результат сохраняется для последующего применения. Метод решения задачи линейного программирования, реализующей рекурсивный алгоритм с перекрывающимися подзадачами, в котором каждая такая подзадача решается один раз, а ее результат сохраняется для последующего применения. | 1 | 3 |
|  | Какую задачу нецелесообразно решать при помощи динамического программирования: | RadioButton | линейного программирования; о рюкзаке; о расстановке скобок при умножении матрицы; вычисления дистанции Левенштейна; | 1 | 3 |
|  | При решении задачи вычисления дистанции Левенштейна с одинаковыми исходными данными при помощи динамического программирования и рекурсивных алгоритмов какой из методов решения является более быстрым: | RadioButton | задача вычисления дистанции Левенштейна не решается этими методами; рекурсивные алгоритмы; примерно равны; динамического программирования; | 4 | 3 |
|  | К оптимизационным алгоритмам на графах не относятся: | RadioButton | алгоритм поиска в ширину; алгоритм поиска в высоту; алгоритм поиска в глубину; топологическая сортировка; | 2 | 3 |
|  | Алгоритм поиска в ширину заключается в: | RadioButton | 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; упорядочивания вершин безконтурного ориентированного графа согласно линейного порядка; посещении вершин в порядке их удаленности от некоторой заранее выбранной или указанной стартовой вершины; выборе вершин случайным образом; | 3 | 4 |
|  | Алгоритм поиска в глубину заключается в: | RadioButton | посещении вершин в порядке их удаленности от некоторой заранее выбранной или указанной стартовой вершины; упорядочивания вершин безконтурного ориентированного графа согласно линейного порядка; выборе вершин случайным образом; 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; | 4 | 4 |
|  | Алгоритм топологической сортировки заключается в: | RadioButton | упорядочивания вершин безконтурного ориентированного графа согласно линейного порядка; 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; посещении вершин в порядке их удаленности от некоторой заранее выбранной или указанной стартовой вершины; выборе вершин случайным образом; | 1 | 4 |
|  | Упрощенная задача о рюкзакевыглядит следующим образом: Существует n различных предметов, характеризующихся объемом v и стоимостью vc. Необходимо выбрать несколько разных предметов таким способом, чтобы они поместились в рюкзаке объемом Vи при этом их суммарная стоимость была максимальной. Решением задачи при такой постановке будет вектор (x1,x2,…,xn). Каждый элементxi вектора может принимать значение или 1. При этом если xi=0 то i-ый предмет не выбран, и если xi = 1, то i-й предмет выбран для размещения в рюкзаке. Как может быть записана математическая модель задачи: | RadioButton | <http://portal.by/uploads/2c81619662652554cf70f80c960d4842.jpg> <http://portal.by/uploads/c8a41a5d44cedbe618f307b0f68bd28a.jpg><http://portal.by/uploads/b29195f0b5ffcfa5b13f1f75ef96ffd1.jpg> <http://portal.by/uploads/5c3ff56521b382a835368f450b0c6f45.jpg> | 2 | 5 |
|  | Упрощенная задача о рюкзаке имеет следующие входные данные:V=100 – вместимость (объем) рюкзака;n = 4 – количество предметов; (25,30,60,20) – вектор объемов предметов; (25,10,20,30) – вектор стоимостей предметов. На рисунке представлена схема решения. Предметы с каким номером должны быть помещены в рюкзак?[http://portal.by/uploads/5cc0e341ac867debec0f936e3d44ef61.JPGhttp://portal.by/uploads/5cc0e341ac867debec0f936e3d44ef61.JPG](http://portal.by/uploads/5cc0e341ac867debec0f936e3d44ef61.JPG) | RadioButton | 4; 3, 2 и 1; 4, 2 и 1; 3 и 1; | 4 | 4 |
|  | 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): | RadioButton | <http://portal.by/uploads/bc1f2ebadc60c4afc01d596c1e818cb6.jpg> <http://portal.by/uploads/27bb010c4d93f4b4e1798a413ad03b98.jpg><http://portal.by/uploads/141097546fc4d67d9634fd9cb843cf7d.jpg> <http://portal.by/uploads/0e5bbfa71c05587691dbf709e843305a.jpg> | 2 | 5 |
|  | 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? <http://portal.by/uploads/12352b21b1bac74d9171f1d13d7d92ae.jpg> | RadioButton | 18; 19; 10; 15; | 1 | 4 |
|  | Ниже представлен фрагмент программного кода для реализации задачи о загрузке судна.  // --- Main #include "stdafx.h" #include <iostream> #include <iomanip> #include "Boat.h" #define NN (sizeof(v)/sizeof(int)) #define MM 3 int \_tmain(int argc, \_TCHAR\* argv[]) { setlocale(LC\_ALL, "rus"); int V = 1000,  v[] = {100, 200, 300, 400, 500, 150},  c[NN] = { 10, 15, 20, 25, 30, 25}; short r[MM]; int cc = boat( V, // [in] максимальный вес груза MM, // [in] количество мест для контейнеров NN, // [in] всего контейнеров  v, // [in] вес каждого контейнера  c, // [in] доход от перевозки каждого контейнера r // [out] результат: индексы выбранных контейнеров ); std::cout<<std::endl<<"- Задача о размещении контейнеров на судне";  std::cout<<std::endl<<"- общее количество контейнеров : "<< NN; std::cout<<std::endl<<"- количество мест для контейнеров : "<< MM; std::cout<<std::endl<<"- ограничение по суммарному весу : "<< V; std::cout<<std::endl<<"- весконтейнеров : "; for(int i = 0; i < NN; i++) std::cout<<std::setw(3)<<v[i]<<" "; std::cout<<std::endl<<"- доходотперевозки : "; for(int i = 0; i < NN; i++) std::cout<<std::setw(3)<<c[i]<<" "; std::cout<<std::endl<<"- выбраныконтейнеры (0,1,...,m-1): ";  for(int i = 0; i < MM; i++) std::cout<<r[i]<<" "; std::cout<<std::endl<<"- доход от перевозки : " << cc; std::cout<<std::endl<<"- общийвесвыбранныхконтейнеров : "; int s = 0; for(int i = 0; i < MM; i++) s+= v[r[i]]; std::cout<<s;  std::cout<<std::endl<<std::endl; system("pause"); return 0; } | RadioButton | <http://portal.by/uploads/a3c7eb95d75982bc1dff7bf707cf217f.jpg> <http://portal.by/uploads/cf4a5166905350b27e79f89664d3e3bd.jpg><http://portal.by/uploads/42190028419c04aa71f4c8f4cb0e5fc4.jpg> <http://portal.by/uploads/3190405d4d48a044e9c72f1d0b5966cf.jpg> | 4 | 2 |
|  | 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)? | RadioButton | <http://portal.by/uploads/b59a2c0624b70469f0bf5e1fba4f0084.jpg> <http://portal.by/uploads/ad6f53d49ada40c18d07fe3ebf9a0671.jpg><http://portal.by/uploads/df0e41ba51f12f52f448ead3815d7a37.jpg> <http://portal.by/uploads/691ee82b092898567e881b65499b45e8.jpg> | 1 | 5 |
|  | На рисунке изображена схема решения задачи коммивояжера. Задача решается для пяти городов. Расстояние между городами задается матрицей A. Какой маршрут будет являться решением? <http://portal.by/uploads/c6a054ac5cd43875447726f8ee35e019.jpg><http://portal.by/uploads/f38cd7a899351301860178ba0258f640.jpg> | RadioButton | 0-4-2-1-3-0; 0-3-4-2-1-0; 0-3-1-4-2-0; 0-2-4-3-1-0; | 2 | 4 |
|  | 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? // -- main  #include "stdafx.h" #include "Auxil.h" #include <iostream> #include <iomanip> #include <time.h> #include "Salesman.h" #define SPACE(n) std::setw(n)<<" " #define N 12 int \_tmain(int argc, \_TCHAR\* argv[]) { setlocale(LC\_ALL, "rus"); int d[N\*N+1], r[N];  auxil::start();  for(int i = 0; i <= N\*N; i++) d [i] = auxil::iget(10,100);  std::cout<<std::endl<<"-- Задача коммивояжера -- "; std::cout<<std::endl<<"-- количество ------ продолжительность -- "; std::cout<<std::endl<<" городов вычисления "; clock\_t t1, t2;  for (int i = 7; i <= N; i++) {  t1 = clock(); salesman (i, (int\*)d, r);  t2 = clock(); std::cout<<std::endl<<SPACE(7)<<std::setw(2)<<i <<SPACE(15)<<std::setw(5)<<(t2-t1); } std::cout<<std::endl; system("pause"); return 0; }  <http://portal.by/uploads/2ad2179e56cedf99daaad016195c455a.jpg> | RadioButton | минуты; секунды; условные единицы процессорного времени; сутки; | 3 | 4 |
|  | Условие задачи о загрузке судна с условием центровки звучит следующим образом: На палубе судна имеется m мест для размещения грузовых контейнеров. Выбрать m контейнеров для погрузки на судно можно из n больше либо равно m имеющихся в наличии. Каждый контейнер i характеризуется весом v и доходом c от его перевозки. На каждое место можно разместить контейнер, если его вес не превышает заданные величины. Необходимо выбрать m контейнеров из n имеющихся таким образом, чтобы доход от перевозки был максимально возможным. Как может быть записана математическая модель задачи (ki – неизвестные (номера выбранных контейнеров), которые требуется найти)? | RadioButton | <http://portal.by/uploads/baf8ad1b81c1b78fa183803c514a6539.jpg> <http://portal.by/uploads/163bd5a960d45808522aaa1f31f5027c.jpg><http://portal.by/uploads/656b768c559775e0465158e8ee193283.jpg> <http://portal.by/uploads/5f143966d9ee79342cc6d557330b499c.jpg> | 4 | 5 |
|  | 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 судна с условием центровки?<http://portal.by/uploads/5b5a33fab556d68f13df03626f70d5ad.jpg> | RadioButton | 5; 16; 20; 22; | 4 | 5 |
|  | Граф решения какой задачи представлен рисунке: <http://portal.by/uploads/5c7cf038f215af13164ee3510e9b4e64.jpg> | RadioButton | о коммивояжере; о рюкзаке; о загрузке судна; о загрузке судна с центровкой; | 1 | 3 |
|  | На рисунке представлено изображение некого графа. Данный граф является:<http://portal.by/uploads/9d403f9fe73208cdb1db418dd65aa03a.jpg> | RadioButton | неориентированным; ориентированным; взвешенным; изображение, представленное на рисунке, не является графом; | 2 | 2 |
|  | Теорема Форда-Фалкерсона: | RadioButton | В любой сети максимальная величина потока из истока s в сток t равна максимальной пропускной способности разреза отделяющего s от t; В любой сети максимальная величина потока из истока s в сток t равна 2(s+t); В любой сети максимальная величина потока из истока s в сток t равна минимальной пропускной способности разреза отделяющего s от t; В любой сети максимальная величина потока из истока s в сток t равна 2(s–t); | 3 | 5 |
|  | Назначение транспортной задачи: | RadioButton | определение количества автомобилей для доставки грузов из пунктов отправления в пункты назначения; определение количества требуемого объема топлива для автомобилей для доставки грузов из пунктов отправления в пункты назначения; определение типа транспорта (авто, ж/д, авиа и т.д.) для доставки грузов из пунктов отправления в пункты назначения; определение объемов перевозок из пунктов отправления в пункты назначения с минимальной суммарной стоимостью перевозок; | 4 | 3 |
|  | 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: | RadioButton | <http://portal.by/uploads/e46ae62d1f9032cfcebbe75720b25dac.jpg> <http://portal.by/uploads/efdd65820fb302656b78bec38f684a65.jpg><http://portal.by/uploads/69fbbb7a6f810bbb34099bf0e4018538.jpg> <http://portal.by/uploads/7ea51b8fb0c1bcce6b8f2506b3a28c30.jpg> | 4 | 4 |
|  | Из скольких этапов состоит решение транспортной задачи: | RadioButton | 1 2 3 4 | - | 2 |
|  | Одним из возможных определений понятия линейное программирование является: | RadioButton | раздел программирования, посвященный теории и методам решения экстремальных задач на языке С++; область математического программирования, посвященная теории и методам решения экстремальных задач, характеризующихся нелинейной зависимостью между переменными; раздел программирования, посвященный теории и методам решения экстремальных задач на языке Java; область математического программирования, посвященная теории и методам решения экстремальных задач, характеризующихся линейной зависимостью между переменными; | 4 | 3 |
|  | Имеется некая задача линейного программирования с двумя неизвестными. Какими методами следует решать данную задачу: | RadioButton | графическим и симплекс-методом; графическим и методом случайного поиска; симплекс-методом и методом случайного поиска; графическим, симплекс-методом и методом случайного поиска; | 1 | 4 |
|  | Одним из возможных определений понятия нелинейное программирование является: | RadioButton | область математического программирования, посвященная теории и методам решения экстремальных задач, характеризующихся линейной зависимостью между переменными; раздел программирования, посвященный теории и методам решения экстремальных задач на языке С++; область математического программирования, посвященная теории и методам решения экстремальных задач, характеризующихся нелинейной зависимостью между переменными; раздел программирования, посвященный теории и методам решения экстремальных задач на языке Java; | 3 | 3 |
|  | 3) 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)): | RadioButton | Z(x) стремится к (min,max) Z(x)=X Z(x)=min Z(x) стремится к max | 1 | 3 |