**Тема 1 Теория NP**

Можно ли решить NP-полные задачи за полиномиальное время?

Нельзя решить, можно свести другую NP задачу к NP-полной за полиномиальное время.

Можно ли решить задачи класса Р за полиномиальное время?

P-это класс сложности, представляющий собой совокупность всех задач решения, которые могут быть решены за полиномиальное время.

Можно ли решить задачи класса NР за полиномиальное время?

NP-это класс сложности, представляющий собой набор всех задач решения, для которых случаиимеют доказательства, которые могут быть проверены за полиномиальное время.

Можно ли найти приближенное решение для задачи класса NР (хотя бы для одной задачи из этого класса) за полиномиальное время?

Да, Разработаны приближенные эффективные алгоритмы с гарантированными оценками точности для ряда NP-трудных задач

Можно ли найти приближенное решение для NР-трудной задачи за полиномиальное время (хотя бы для какой-нибудь одной задачи из этого класса)?

Да, Разработаны приближенные эффективные алгоритмы с гарантированными оценками точности для ряда NP-трудных задач

Можно ли найти приближенное решение для NР-полной задачи за полиномиальное время (хотя бы для какой-нибудь одной задачи из этого класса)?

Нет.

Существуют ли задачи, которые не могут быть решены никаким алгоритмом?

Да, неразрешимые задачи

Существуют задачи, для которых существует решающий их алгоритм, но любой такой алгоритм работает долго – время его работы не есть О(*nk*) ни для какого фиксированного числа k?

Да существуют задачи класса NP

В чем состоит в теории алгоритмов проблема, называемая «Р ≠ NP»?

Если положительный ответ на какой-то вопрос можно быстро (за полиномиальное время) проверить (используя некоторую вспомогательную информацию, называемую сертификатом), то верно ли, что и сам ответ (вместе с сертификатом) на этот вопрос можно быстро найти? Задачи первого типа относятся к классу P, второго — к классу NP. ???

В чем состоит *интересное* свойство NP-полного класса задач?

1. Никакую NP-полную задачу нельзя решить никаким известным полиномиальным алгоритмом, несмотря на настойчивые усилия многих блестящих исследователей.

2. Если бы удалось построить полиномиальный алгоритм для какой-нибудь NP-полной задачи, то это бы означало полиномиальную разрешимость каждой NP-полной задачи.

3. Задача должна принадлежать классу NP (L  NP)

4. К ней полиномиально должны сводиться все задачи из класса NP (Lx ≤P L, для каждого Lx NP)

5. Если существует задача, принадлежащая классу NPC, для которой существует полиномиальный алгоритм решения F = O(n k ), то класс P совпадает с классом NP, т.е. P = NP .

Какое интересное хз

Что означает понятие – «*Класс задач Р, состоящий из «задач», разрешимых за полиномиальное время*»

существует константа k и алгоритм, решающий задачу с (n)=O(http://th-algoritmov.narod.ru/img/nk.gif), где n - длина входа алгоритма в битах n = |D|

Укажите свойства P класса задач?

* для большинства задач из класса P константа k меньше 6;
* класс P инвариантен по модели вычислений (для широкого класса моделей);
* класс P обладает свойством естественной замкнутости (сумма или произведение полиномов есть полином).

Задача (проблема, язык) *L* {0, 1}\* называется **NP-полной** (NPC), если

1) L∈ NP;

2) L′ ≤P L для любого L′ ∈ NP.

Задача (проблема, язык) *L* {0, 1}\* называется **NP-трудной** (NPН), если… (*Закончите фразу*

Язык L называется NP-трудным (NPH), если для любого языка L1, принадлежащего NP, L1 сводится по Карпу к L (L1≤L).

Укажите основное свойство NP-полных задач

1. Никакую NP-полную задачу нельзя решить никаким известным полиномиальным алгоритмом, несмотря на настойчивые усилия многих блестящих исследователей.

2. Если бы удалось построить полиномиальный алгоритм для какой-нибудь NP-полной задачи, то это бы означало полиномиальную разрешимость каждой NP-полной задачи.

полиномиальная сводимость

Какое основноехз

Если для задачи (проблемы, языка) L найдется задача (проблема, язык) L'*∈*NPC, для которого L'*<=p*L, то задача L NP-полная

Всякий ли NP-полный язык является NP-трудным?

Да

Всякий ли NP-трудный язык является NP-полным?

Нет

Что требуется, чтобы NP-трудная задача являлась и NP-полной?

Ее решение можно проверить за полиномиальное время, то она должна принадлежать к NP

**Тема 2 Асимптотическая оценка трудоемкости кода программы**

Определите асимптотическую оценку O(*f*(*N*)) функции роста трудоемкости следующего фрагмента программы: int F(A[][][],n) {

inti, j, k, s=0;

for(i=**1**; i<= **n**; i**+=2**)

for(j=**1**; j<= **n-1**; j**\*=3**)

for(k=**1**; k<= **n-2**; **k++**)

s=s+A[i][j][k];

returns;

}

**Тема 3Рекурсия**

Чему равен ***порядок*** рекуррентного соотношения

Порядок равен 2

Укажите формулу, являющуюся решением**(**ПРИМЕРОМ решения**)** некоторого рекуррентного уравнения:

*Примерные варианты ответов, определите верный (верные):*

?

?

?

Решением рекуррентного уравнения может быть… (укажите допустимые формы записи решения рекуррентного уравнения)

f(1) =3

В ходе метода **прямой** подстановки выполняются следующие вычисления (укажите допустимые отдельные шаги, этапы, записи выражений, применяемых в ходе прямой подстановки):

F (1) = 1

F (2) =2\*F (2-1) + 1 = 2\*1 + 1 = 3

F(n) =2\*F(n-1)+1

В ходе метода **обратной** подстановки выполняются следующие вычисления (укажите допустимые отдельные шаги, этапы, записи выражений, применяемых в ходе обратной подстановки)

х(n) = х(n-1)+n.

x(n-1) = x(n-2)+n-1

x(n) = x(n-i) + x(n-i+1) + x(n-i+2)+n

В ходе метода обратной подстановки выполняются следующие вычисления:

*Примерные варианты ответов, определите верный (верные):*

?

?

Данное дерево рекурсии может соответствовать рекуррентному выражению:

*Примерные варианты ответов:*

–>

–>

**Тема 4Асимптотика вложенности сумм**

Определите формулы, которые приводятся касимптотической оценке***O(f(N)) = n***

*Примерные варианты ответов, определите верные:*

–>

–>,*k=const*

–>

–>

**Тема 5Соответствие асимптотической оценки трудоемкости кода программы**

Определите фрагмент программы, соответствующий асимптотической оценке O(*f*(*N*))=n3:

*Примерные варианты ответов, определите верный (верные):*

? int F(A[][][],n) {int i, j, k, s=0; for(i=**1**; i<=**n\*n**; **i++**) for(j=**1**; j<=**n-1**; **j++**);}

? int F(A[][][],n) {int i, j, k, s=0; for(i=**1**; i<=**2\*n**; **i++**) for(j=**1**; j<=**n-1**; **j++**);}

? int F(A[][][],n) {int i, j, k, s=0; for(i=**1**; i<=**n**; **i++**) for(j=**1**; j<=**n-1**; **j++**); for(k=**1**; j<=**n-1**; **k++**);}

**Тема 6. Основные понятия Анализа сложности алгоритма и программы**

Величина, отражающая порядок величины требуемого ресурса (времени или дополнительной памяти) в зависимости от размерности задачи – это Сложность алгоритма (закончите определение)

Сложность алгоритма бывает

1 временная T(n)

2 Пространственная(ёмкостная) V(n)

??(какой термин может быть применим в теории ресурсной эффективности)

На время выполнения программы влияет … (закончите фразу)

1 исходные данные d(n), от их “размера”

2 F(N)- Время выполнения алгоритма

Время выполнения программы измеряется в

физических единицах времени (секунды, миллисекунды, «тики» ЦП)(закончите фразу)

Время выполнения алгоритма измеряетсяв "шагах" (инструкциях алгоритма) (закончите фразу)

Времени работы **алгоритма***Tmax*(*n*) соответствует сложности наиболее

неблагоприятного случая, когда алгоритм работает дольше всего (закончите фразу)

Времени работы **алгоритма***Tmid*(*n*) (или *Tсред*(*n*) )соответствует сложности алгоритма в среднем (закончите фразу)

Времени работы **алгоритма***Tmin*(*n*)соответствует сложности в наиболее

благоприятном случае, когда алгоритм справляется быстрее всего (закончите фразу)

Времени работы **программы***Tmax*(*n*) соответствует … (закончите фразу)

Времени работы **программы***Tmid*(*n*) (или *Tсред*(*n*)) соответствует … (закончите фразу)

Времени работы **программы***Tmin*(*n*)соответствует … (закончите фразу)

**Тема 7. Основные понятия аппарата О-символики**

Обозначение O(*g*(*n*) – это означает, что g(n) является асимптотически точной оценкой(закончите фразу, объясните суть термина­­ – выберите верные суждения)

Про запись f(*n*) = O(*g*(*n*))можно сказать, что g(n) является асимптотически точной оценкой для f(n)(закончите фразу, объясните суть термина­­ – выберите верные суждения)

Если для функции *T*(*n*) найдутся константы *c*1> 0 и *c*2> 0 и такое число *n*0> 0, что выполнится условие *c*1*n*2 ≤ *T*(*n*) ≤ *c*2*n*2, то T(n) = Θ(n^2)(закончите фразу)

Если для функций *X*(*n*) >0 и *Y*(*n*) >0 найдутся константы *c*1> 0 и *c*2> 0 и такое число *n*0> 0, что выполнится условие *c*1*Y(n)*≤ *X*(*n*) ≤ *c*2*Y(n)*, то X(n) = Θ(Y(n))(закончите фразу)

Если для функций *X*(*n*) >0 и *Y*(*n*) >0 найдется константа *c*1> 0 и такое число *n*0> 0, что выполнится условие 0 <*c*1*Y(n)*≤ *X*(*n*), то X(n) = Ω(Y(n))(закончите фразу)

Если для функций *X*(*n*) >0 и *Y*(*n*) >0 найдется константа *c*1> 0 и такое число *n*0> 0, что выполнится условие 0≤ *X*(*n*) ≤ *c*1*Y*(*n*), то X(n) = O(Y(n))(закончите фразу)

Если для двух функций *X*(*n*) >0 и *Y*(*n*) >0 установлено, что *X*(*n*) = Θ(*Y*(*n*)). то найдутся константы *c*1> 0 и *c*2> 0 и такое число *n*0> 0, что выполнится условие *c*1*Y(n)*≤ *X*(*n*) ≤ *c*2*Y(n)*(закончите фразу, выбрав верные возможные суждения)

Если для двух функций *X*(*n*) >0 и *Y*(*n*) >0 установлено, что *X*(*n*) = О(*Y*(*n*)). то найдется константа c1 > 0 и такое число n0 > 0, что выполнится условие 0 ≤ X(n) ≤ c1Y(n)(закончите фразу, выбрав верные возможные суждения)

Если для двух функций *X*(*n*) >0 и *Y*(*n*) >0 установлено, что *X*(*n*) = Ω (*Y*(*n*)). то найдется константа *c*1> 0 и такое число *n*0> 0, что выполнится условие 0 <*c*1*Y(n)*≤ *X*(*n*)(закончите фразу, выбрав верные возможные суждения)

Если для двух функций *X*(*n*) >0 и *Y*(*n*) >0 установлено, что *X*(*n*) = o (*Y*(*n*)). то для всякого c> 0 найдется такое n0что 0 ≤ X(n) ≤ cY(n)(закончите фразу, выбрав верные возможные суждения)

Если для двух функций *X*(*n*) >0 и *Y*(*n*) >0 установлено, что *X*(*n*) = ω (*Y*(*n*)). то для всякого c> 0 найдется такое n0 что 0 ≤ cY(n) ≤ X(n)(закончите фразу, выбрав верные возможные суждения)

**Тема 8. Свойства О-символики (транзитивность, рефлексивность, симметричность, обратимость)**

Отметьте для функций *X*(*n*) >0, *Y*(*n*) >0 и *Z*(*n*)>0 строки ответа, соответствующие свойствутранзитивности аппарата О-символики.

??Пусть определено, что *X*(*n*)(n5)= ω(*n*4). Укажите верное соотношение между функциями с помощью О-символики, в том числе с учетом свойства транзитивности:

*Примерные варианты ответов, определите верный (верные):*

*? X*(*n*) = o(*n3*)

*? X*(*n*) = ω(*n3*)

*? X*(*n*)= O(*n3*)

*? X*(*n*) = ω (*n7*)

*? X*(*n*) = Ω (*n4*)

Укажите верные соотношения:

*? X*(*n*) = o(*X*(*n*))

*? n3* = o(*n3*)

*?n5* = Θ (*n5*)

*?*1= O(1)

*? Y*(*n*) = O (*X*(*n*)) ⬄*X*(*n*) = O (*Y*(*n*))

*?Y*(*n*) = Θ (*X*(*n*)) ⬄*X*(*n*) = Θ (*Y*(*n*))

*?Y*(*n*) = o (*X*(*n*)) ⬄*X*(*n*) = ω (*Y*(*n*))

**Тема 9** **Сложение и умножение в O-символике**

Если число тактов (действий), необходимое для работы алгоритма, выражается как 25*n*2 + 10*n*\**log* *n* + 5*n* + 15, то это алгоритм, для которого функция трудоемкости имеет порядок n2, то есть имеет квадратичную зависимость от размера входных данных(закончите фразу, объясните суть термина­­ – выберите верные суждения)

Если *X*1(*n*) = O(*Y*1(*n*)) и *X*2(*n*) = O(*Y*2(*n*)), то чему ***может быть***равным выражение *X*1(*n*) +*X*2(*n*) в О-символике?

*O(max{Y*1(*n*) + *Y*2(*n*)}*)*

Если *X*1(*n*) = O(*Y*1(*n*)) и *X*2(*n*) = O(*Y*2(*n*)), то чему ***может быть*** равным выражение *X*1(*n*) \* *X*2(*n*) в О-символике?

*O(Y*1(*n*)\**Y*2(*n*)*)*

Укажите верные соотношения:

? 2*n*2 + 5*n*= O(*n*3)

? 2*n*2 + 5*n*= Ω(*n*3)

? 2*n*2 + 5*n*= ω(*n*)

? 2*n*2 + 5*n*= o(*n*3)

?(2*n* + 5)2= O(*n*2)

? (2*n* – 1)3**= Θ(*n*3)**

**?**2*n*2log(*n*)+ 5*n*2= Θ(*n*2)

? 2*n*2log(*n*)+ 5*n*2= Θ(*n*2 log(*n*))

?4*log2*(*n*) = Θ(*n*2)

?9*log3*(2*n*) = o(n3)

**Тема 11 Принципы анализа алгоритмов и программ: базовые операции**

К базовым (элементарным) операциям, трудоемкость которых определяется как O(1), относятся:(выберите верные варианты ответов)

– присваивания (=, +=, -=, \*=, /=, %=)

– арифметические операции (+, –, \*, /, % (получение остатка от деления));

– операции сравнения (>, <, ==, !=, <=, >= );

– логические операции (!,&&, ||);

– бинарные операции (~, &, |, ^);

– взятие (получение, использование) адреса ячейки памяти: [] – (в массиве),

\* (разыменовывание), & (получение адреса), -> (указатель на структуру), .

(разделитель имени структуры и поля;

– вызов и выполнение стандартных библиотечных функций (fabs(), scanf(),

printf(),…) количество операций в которых не зависит от входного параметра

**Тема 12 Принципы анализа алгоритмов и программ:**

Трудоемкость выполнения линейной последовательности операторов определяетсясуммой трудоемкостей блоков последовательности(закончите фразу, объясните суть термина­­ – выберите верные суждения)

Трудоемкость вычисления выражения (в круглых скобках) определяетсясуммой трудоемкостей каждой операции в скобках (закончите фразу, объясните суть термина­­ – выберите верные суждения)

Трудоемкость выполнения оператора условия if определяетсявременем вычисления самого логического выражения и

временем выполнения условно исполняемых операторов(закончите фразу, объясните суть термина­­ – выберите верные суждения)

Трудоемкость выполнения оператора выбора switch определяется … (закончите фразу, объясните суть термина­­ – выберите верные суждения)

• в операторе выбора switch анализируется каждая case ветвь, включая ветвь default трудоемкость каждой определяем отдельно, а затем выбираем ту ветвь, функция роста трудоемкости которой имеет бОльший порядок.

Укажите факторы, которые надо учитывать при определении трудоемкости выполнения оператора цикла for(;;)трудоемкость тела цикла, трудоемкость итераторов, проверки условия.(закончите фразу, объясните суть термина­­ – выберите верные суждения)

Трудоемкость цикла for(;;) вычисляетсясуммой времени всех исполняемых итераций цикла, в свою очередь состоящих из времени выполнения операторов тела цикла и времени вычисления условия прекращения цикла(закончите фразу, объясните суть термина­­ – выберите верные суждения)