1. Wszystkie problemy optymalizacji kombinatorycznej, które dają się rozwiązać do optymalności w wielomianowym czasie:
   a. muszą być matroidami
   b. muszą być z klasy NP
   c. mogą być matroidami
   d. muszą być z klasy P

2. Jeżeli P=NP to
   a. wszystkie problemy NP-trudne dadzą się rozwiązać w wielomianowym czasie
   b. żadne problemy NP-równoważne nie dadzą si rozwiązać w wielomianowym czasie
   c. wszystkie problemy NP-równoważne dadzą się rozwiązać w wielomianowym czasie
   d. żadne problemy NP-łatwe nie dadzą się rozwiązać w wielomianowym czasie

3. Jeżeli P nie jest równe NP (P!=NP) to
a. wszystkie problemy NP-łatwe dadzą się rozwiązać w wielomianowym czasie
b. wszystkie problemy NP-trudne dadzą się rozwiązać w wielomianowym czasie
c. żadne problemy NP-łatwe nie dadzą się rozwiązać w wielomianowym czasie
d. żadne problemy NP-trudne nie dadzą się rozwiązać w wielomianowym czasie

4. Wyobraźmy sobie, że dla problemu szeregowania zadań na równoległych identycznych procesorach dla kryterium długości uszeregowania (P||Cmax) podano wielomianowe algorytmy aproksymacyjne A,B spełniające dla każdej instancji I warunki 1) algorytm: A: |A(I)-OPT(I)|<=k, gdzie k>0 jest stałą, 2) algorytm B: B(I)/OPT(I)<=k, gdzie 1<k? Wynika z tego, że:
   a. z 1) wynika, że P=NP, z 2) wynika, że P!=NP
   b. z 1) wynika, że P=NP, z 2) nic nie wynika
   c. z 1) nic nie wynika, z 2) wynika, że P=N
   d. z 1) wynika, że P!=NP, z 2) wynika, że P=NP

Theorem

If $P \neq NP$, then no polynomial algorithm A for knapsack can guarantee that $|A(I) - OPT(I)| \leq k$, where $k > 0$ is constant.

5. Problem optymalizacji kombinatorycznej jest nieaproksymowalny gdy:

a. jest NP-trudny
b. nie jest znany dla niego wielomianowy algorytm o skończonym oszacowaniu jakości
c. można wykazać, że nie istnieje dla niego wielomianowy algorytm o skończonym oszacowaniu
d. istnieje dla niego wielomianowy algorytm o skończonym oszacowaniu jakości
e. istnienie dla niego wielomianowego algorytmu o gwarantowanej jakości pociągałoby P=NP
f. ogólnie, gdy P jest różne od NP
g. ten problem nie transformuje się wielomianową transformacją Turinga do innego aproksymowalnego problemu optymalizacji kombinatorycznej.
h. istnienie dla niego wielomianowego algorytmu o gwarantowanej jakości pociągałoby P!=NP.

b) nie ponieważ nie jest znany != nie istnieje (c is better)
e) is sus

6. Przetarg między jakością rozwiązań i czasem wykonania w poprawnie skonstruowanym algorytmie dla problemu optymalizacji kombinatorycznej oznacza, że:

a. im krótszy czas działania tym lepszej jakości rozwiązania
b. dobrej jakości rozwiązanie są łatwe do uzyskania w krótkim czasie
c. im dłuższy czas działania tym lepszej jakości rozwiązania
d. z upływem czasu jakość rozwiązania się ustala

7. W kolejnych iteracjach algorytmu Dinica kolejne sieci warstwowe mają

a. stałą liczbę warstw
b. niemalejącą liczbę warstw
c. rosnącą liczbę warstw
d. liczba warstw nie ma związku z numerem iteracji w algorytmie Dinica

8. Jeżeli w grafie istnieje cykl o ujemnej długości, to

a. między niektórymi wierzchołkami da się, a między innymi nie da się określić odległości
b. wszystkie najkrótsze ścieżki w grafie przechodzą przez ten cykl
c. między wszystkim wierzchołkami da się określić odległość
d. między żadnymi wierzchołkami nie da się określić odległości

- Conclusion: if there is a negative-length cycle is $G(V, A)$ then distance between some nodes may be *undefined* (depends if a cycle is accessible and how many times one traverses the cycle).

9. Porównując algorytmy zachłanne i dokładne dla problemów optymalizacji kombinatorycznej

a. tylko algorytmy dokładne dają gwarancję optymalności rozwiązań
b. tylko algorytmy zachłanne dają gwarancję optymalności
c. ani algorytmy dokładne ani zachłanne nie dają gwarancji optymalności rozwiązań.
d. algorytmy dokładne i zachłanne dają gwarancję optymalności rozwiązań

e. Wykładnicze są algorytmy dokładne i zachłanne

f. Wielomianowe są algorytmy dokładne, zachłanne są wykładnicze.

g. Wykładnicze są algorytmy dokładne, zachłanne są wielomianowe

e. Algorytmy dokładne i zachłanne są wielomianowe

a is sus (istnieją algorytmy zachłanne zwracające rozwiązanie optymalne)

- Exact algorithms have exponential runtime in the worst case.

10. Wyobraźmy sobie, że 1) dla problemu wierzchołkowego kolorowania grafu podano wielomianowy algorytm A dla każdej instancji I spełniający warunek A(I)/OPT(I)<4/3, 2) dla problemu krawędziowego kolorowania grafu podano wielomianowy algorytm B spełniający warunek: |B(I)-OPT(I)|≤k. Gdzie 0<k<2 jest stałą, A(I), B(I) oznaczają jakość rozwiązań aproksymacyjnych, OPT(I) wartość optymalna. Z tego wynika, że:

a. z 1) wynika, że P=NP, z 2) wynika że P!=NP
b. z 1) wynika, że P!=NP, z 2) wynika że P=NP
c. z 1) wynika, że P=NP, z 2) nic nie wynika
d. z 1) nic nie wynika, z 2) wynika, że P=NP.

Algorytm pierwszy zwraca zawsze optymalne (można podstawić i sprawdzić)

11. W świetle obecnego stanu wiedzy pewne problemy optymalizacji kombinatorycznej 1) nie dają się rozwiązać do optymalności algorytmami zachłannymi, 2) inne dają się rozwiązać do optymalności algorytmami zachłannymi. Jest tak gdyż:

a) problemy 1) są matroidami, problemy 2) są z klasy P
b) problemy 1) są z klasy P, problemy 2) są matroidami
c) problemy 1) są NP-trudne, problemy 2) są matroidami
d) problemy 1) są NP-zupełne, problemy 2) są z klasy P

Jeżeli coś jest matroidem ważonym to musi istnieć dla niego algorytm zachłanny do optymalności.

12. W problemie wyznaczania maksymalnego przepływu w sieci, ścieżka powiększająca przepływ zawiera 1) łuki zwrócone od s do t i 2) wrócone od t do s, gdy

a. łuki 1) mają przepływ równy pojemności, łuki 2) mają przepływ dodatni
b. łuki 1) mają przepływ dodatni, łuki 2) mają przepływ zerowy
c. łuki 1) mają przepływ mniejszy od pojemności, łuki 2) mają przepływ dodatni
d. łuki 1) mają przepływ dodatni, łuki 2) mają przepływ mniejszy od pojemności

13. W problemie wyznaczania maksymalnego przepływu w sieci, ścieżka powiększająca przepływ

a. może zawierać łuki zwrócone od t do s gdy przepływ na nich jest nieujemny
b. może zawierać łuki zwrócone od t do s gdy przepływ na nich jest dodatni
c. może zawierać tylko łuki zwrócone od s do t
d. może zawierać łuki zwrócone od t do s gdy przepływ na nich jest zerowy

14. W problemie wyznaczania maksymalnego skojarzenia w grafie, chodzi o to, aby

a. wybrać krawędzie, które do siebie nie przylegają w żadnych wierzchołkach
b. wybrać określone wierzchołki z grafu
c. wybrać krawędzie, które tworzą ścieżkę naprzemienną
d. wybrać krawędzie, które do siebie przylegają we wskazanych wierzchołkach

15. Jeżeli dla jakiegoś problemu optymalizacji kombinatorycznej zostanie podany algorytm aproksymacyjny o pewnym oszacowaniu X, to dla lepszego oszacowania możliwości tego algorytmu podaje się jeszcze instancję, dla której rozwiązanie tym algorytmem ma

a. wartość jak najbliższą OPT
b. wartość jak najbliższą X
c. wartość jak najbliższą X*OPT
d. wartość jak najdalszą od X*OPT

????????????????????

16. Problem NP-łatwy jest

a. jest problemem optymalizacyjnym
b. co najwyżej tak trudny jak problemy NP-zupełne
c. co najmniej tak trudny jak problemy NP-zupełne
d. jest problemem decyzyjnym

e. z klasy NP
f. co najmniej tak trudny jak klasa NP
g. rozwiązywalny w wielomianowym czasie
h. co najwyżej tak trudny jak klasa NP

17. Odległość między wierzchołkami w grafie może być nieokreślona, gdy

a. Graf nie jest spójny
b. Graf ma ścieżkę o ujemnej długości
c. Graf ma klikę o ujemnej długości
d. Graf ma cykl o ujemnej długości

Klika to typ grafu (chyba nie istnieje klika ujemnej długości)

18. Odległość między wierzchołkami w grafie jest określona, gdy

a. graf nie ma cyklu o nieujemnej długości
b. graf nie ma cyklu o ujemnej długości
c. graf nie ma ścieżki o ujemnej długości
d. graf jest spójny

19. Jeżeli pewien problem optymalizacji kombinatorycznej jest silnie NP-zupełny w wersji decyzyjnej, to

a. Istnienie w pełni wielomianowego schematu obliczeń pociąga P!=NP
b. Istnienie w pełni wielomianowego schematu obliczeń (FPTAS) pociąga P=NP
c. P=NP pociąga istnienie w pełni wielomianowego schematu obliczeń (FPTAS) imo c
d. P!=NP pociąga istnienie w pełni wielomianowego schematu obliczeń (FPTAS)

Ze slide 72: P!=NP -> sNPh nie mają FPTASes, p->q ⇔ -q -> -p więc odp b

20. Ścieżka powiększa skojarzenie na zasadzie różnicy symetrycznej krawędzi i bieżącego skojarzenia jeżeli

a. jest naprzemienna
b. jest nieparzystej długości i naprzemienna
c. jest nieparzystej długości
d. jest parzystej długości i naprzemienna

21. Najkrótsze ścieżki w grafie są proste ponieważ

a. <u>Wszystkie cykle są nieujemnej długości</u>
b. Wszystkie ścieżki są nieujemnej długości
c. Między wszystkimi parami wierzchołków odległości są nieujemne
d. Między wszystkimi parami wierzchołków odległości są dodatnie

e. Wszystkie cykle są dodatniej długości

f. W grafie są tylko ścieżki dodatniej długości

g. Wszystkie cykle są nieujemnej długości

h. W grafie są tylko ścieżki nieujemnej długości

22. Skojarzenie w grafie jest maksymalne gdy

a. nie istnieją w nim żadne ścieżki zaczynające i kończące się w wierzchołku wolnym
b. nie istnieją w nim ścieżki naprzemienne zaczynające i kończące się w wierzchołku wolnym
c. nie istnieją w nim ścieżki naprzemienne o nieparzystej długości
d. nie istnieją w nim ścieżki nieparzystej długości zaczynające i kończące się w wierzchołku wolnym

e. nie istnieją w nim ścieżki o nieparzystej długości

f. żaden z pozostałych warunków nie jest wystarczający aby skojarzenie było maksymalne

g. nie istnieją w nim ścieżki naprzemienne

h. nie istnieją ścieżki zaczynające i kończące się w wierzchołku wolnym

Skojarzenie w grafie jest maksymalne gdy nie istnieją w nim ścieżki naprzemienne nieparzystej długości zaczynające się i kończące się w wierzchołku wolnym.

23. Jeżeli P jest różne od NP (P!=NP) to problem NP-trudny

a. daje się rozwiązać w wielomianowym czasie
b. nie daje się rozwiązać w wielomianowym czasie
c. to nie jest rozstrzygające

24. W krawędziowym kolorowaniu grafu jeden kolor to zbiór krawędzi, który tworzy
a. Klikę
b. Drzewo
c. Skojarzenie
d. Cykl

25. Trudny problem optymalizacji kombinatorycznej jest aproksymowalny gdy
  a. Istnieje dla niego wielomianowy algorytm o gwarantowanej jakości rozwiązań
  b. Ten problem transformuje się wielomianową transformacją Turinga do innego aproksymowalnego problemu optymalizacji kombinatorycznej
  c. P jest równe NP
  d. Istnieje dla niego jakikolwiek algorytm o gwarantowanej jakości rozwiązań

26. Rozwiązaniem problemu wyboru algorytmu dla optymalizacji kombinatorycznej jest
  a. minimalny czas wykonania algorytmu dla zadanej instancji
  b. najlepsza instancja dla każdego algorytmu
  c. instancja na której dany algorytm wykona się w minimalnym czasie
  d. najlepszy algorytm dla każdej instancji

27. Wyobraźmy sobie, że 1) dla problemu wierzchołkowego kolorowania grafu podano w pełni wielomianowy schemat obliczeń, 2) dla problemu krawędziowego kolorowania grafu podano wielomianowy algorytm B spełniający warunek: |B(I)-OPT(I)|<=k. Gdzie 0<k<2 jest stałą, B(I) oznacza jakość rozwiązań aproksymacyjnych, OPT(I)
  a. Z 1) nic nie wynika, z 2) wynika że P=NP
  b. Z 1) wynika, że P=NP 2) wynika że P!=NP
  c. Z 1) wynika, że P=NP, z 2) nic nie wynika
  d. Z 1) wynika, że P!=NP, z 2) wynika że P=NP

28. Problem sortowania liczb należy do klasy problemów
  a. decyzyjnych
  b. do klasy P
  c. do klasy NP-trudnych
  d. Przeszukiwania

29. Hill climber to
  a. Zaawansowana wersja tabu search
  b. Algorytm zachłanny
  c. Uproszczona wersja tabu search
  d. Algorytm populacyjny

a. Algorytm populacyjny
b. Wersja pochodna od symulowanego wyżarzania
c. Algorytm zachłanny
d. Protoplasta symulowanego wyżarzania

30. W problemie wyznaczania maksymalnego przepływu w sieci

a. należy tylko określić maksymalną wartość przepływu przy źródle
b. należy określić przepływ na każdym łuku
c. trzeba najpierw wyznaczyć wąskie gardło (bottle neck)
d. należy tylko określić maksymalną wartość przepływu przy ujściu

e. przepływy łukowe mogą być nieograniczone

f. przepływy łukowe są ograniczone i nieujemne

g. przepływy łukowe mogą być ujemne

h. przepływy łukowe są nieograniczone i nieujemne

i. Z dwoma wyjątkami suma przepływów wchodzących do wierzchołka zawsze równa się sumie przepływów wychodzących

j. Suma przepływów wchodzących do wierzchołka zawsze równa się sumie przepływów wychodzących

k. Suma przepływów wchodzących do wierzchołka jest niezależna od sumy przepływów wychodzących

l. Suma przepływów wchodzących do wierzchołka jest różna od sumy przepływów wychodzących

31. Wyobraźmy sobie, że dla problemu komiwojażera (TSP) mogą istnieć wielomianowe algorytmy aproksymacyjne A,B spełniające dla każdej instancji I warunki 1) algorytm A: |A(I)-OPT(I)|<=k, gdzie k>0 jest stałą, 2) algorytm B: B(I)/OPT(I) <= k gdzie k> 1 jest stałą? Wynika z tego wynika, że:
a. 1) nic nie wynika, z 2) wynika że P=NP
b. 1) wynika, że P=NP, z 2) wynika że P!=NP
c. 1) wynika, że P=NP, z 2) nic nie wynika
d. 1) wynika, że PSIW=NP, z 2) wynika że P=NP

32. Ścieżka powiększająca skojarzenie powiększa skojarzenie bo
   a. bo kończy się w wierzchołku wolnym
   b. zaczyna się w wierzchołku wolnym
   c. ma nieparzystą liczbę krawędzi
   d. bo zaczyna i kończy się w wierzchołku wolnym

33. Problem NP-trudny jest
   a. co najmniej tak trudny jak problemy z klasy NP
   b. co najwyżej tak trudny jak problemy z klasy NP
   c. tak samo trudny jak cała klasa NP
   d. trudność problemów w klasie NP nie jest rozstrzygająca

   e. z klasy NP.

   f. co najwyżej tak trudny jak problemy NP-zupełne

   g. rozwiązywalny w wielomianowym czasie

   h. co najmniej tak trudny jak problemy NP-zupełne

   i. co najwyżej tak trudny jak klasa NP.

   j. z klasy NP.

   k. co najmniej tak trudny jak klasa NP.

   l. rozwiązywalny w wielomianowym czasie

   m. tak samo trudny jak cała klasa NP.

   n. co najwyżej tak trudny jak problemy z klasy NP.

   o. **co najmniej tak trudny jak problemy z klasy NP**.

   p. trudność problemów w klasie NP nie jest rozstrzygająca

   q. rozwiązywalny w wielomianowym czasie

   u. co najmniej tak trudny jak klasa NP.
   v. z klasy NP.
   w. co najwyżej tak trudny jak klasa NP

34. W problemie wyznaczania maksymalnego przepływu w sieci przepływ netto przez pewien przekrój ze strony źródła na stronę ujścia
a. jest zawsze równa sumarycznej pojemności wszystkich łuków przecinanych przez ten przekrój
b. nie zależy od każdego przekroju
c. zależy od każdego przekroju
d. jest mniejsza lub równa sumarycznej pojemności wszystkich łuków przecinanych przez ten przekrój

35. Problem NP-równoważny jest

a. Tak samo trudny jak cała klasa NP
b. Co najwyżej tak trudny jak problemy z klasy NP
c. Trudność problemów w klasie NP nie jest rozstrzygająca
d. Co najmniej tak trudny jak problemy z klasy NP.
e. Z klasy NP.
f. Tak trudny jak klasa NP
g. Jest tak trudny jak klasa P
h. Rozwiązywalny w wielomianowym czasie
i. Tak trudny jak problemy NP-zupełne
j. Z klasy NP
k. Co najmniej trudny jak problemy NP-trudne
l. Rozwiązywalny w wielomianowym czasie

36. Metaheurystyka to metoda która

a. nie ulepsza znalezionych rozwiązań
b. jest zdefiniowana na meta-poziomie
c. jest jedno-przebiegowa
d. **ulepsza znalezione rozwiązania**

37. Algorytm zachłanny daje optymalne rozwiązanie problemu optymalizacji kombinatorycznej pod warunkiem, że

a. Gdy nie jest NP-zupełny
b. można znaleźć ciąg pośrednich rozwiązań od rozwiązania pustego do dowolnego innego
c. Ciąg pośrednich rozwiązań od rozwiązania pustego do dowolnego innego nie musi istnieć
d. Ten problem jest z klasy P
e. rozwiązanie startowe nie jest puste i jest wstępnie znane
f. ten problem jest z klasy P
g. gdy nie jest NP-zupełny
h. rozwiązanie startowe jest puste

38. W problemie minimalnego drzewa rozpinającego w grafie o N wierzchołkach chodzi o to, żeby

a. wybrać n-1 najtańszych krawędzi
b. wybrać n-1 krawędzi
c. wybrać n-1 krawędzi bez cykli
d. wybrać n-1 najtańszych krawędzi bez cykli
e. połączyć wybrane wierzchołki krawędziami tworzącymi drzewo
f. połączyć wszystkie wierzchołki N najtańszymi krawędziami
g. połączyć wszystkie wierzchołki N-1 najtańszymi krawędziami
h. połączyć wszystkie wierzchołki krawędziami

39. To że jakiś problem optymalizacji kombinatorycznej daje się rozwiązać do optymalności algorytmem zachłannym pociąga za sobą, że
a. rozwiązanie puste nie musi być dopuszczalne i zawsze istnieje ciąg kroków od rozwiązania pustego do innego dopuszczalnego
b. rozwiązanie puste jest dopuszczalne i zawsze istnieje ciąg kroków od rozwiązania pustego do innego dopuszczalnego
c. rozwiązanie puste jest dopuszczalne i nie musi istnieć ciąg kroków od rozwiązania pustego do innego dopuszczalnego
d. rozwiązanie puste nie musi być dopuszczalne i nie musi istnieć ciąg kroków od rozwiązania pustego do innego dopuszczalnego

40. W NP-zupełnym problemie podziału zbioru pyta się, czy zbiór liczb można podzielić na dwa podzbiory o równej sumie dobranych liczb. Z tego wynika, że w problemie pakowania liczb w minimalnej liczbie pudełek o stałej wysokości, nie istnieje
- a. algorytm wielomianowy o asymptotycznym oszacowaniu lepszym niż 1.5
- b. algorytm wielomianowy o asymptotycznym oszacowaniu lepszym niż 2
- **c. FPTAS (w pełni wielomianowy schemat obliczeń) dla tego problemu**
- d. żaden wielomianowy algorytm aproksymacyjny

41. O pewnym problemie optymalizacji kombinatorycznej wiadomo, że odpowiedź na pytanie czy istnieje rozwiązanie o wartości X>0 jest problemem NP-zupełnym. Z tego wynika, że
a. nie mogą istnieć wielomianowe algorytmy aproksymacyjne o skończonym oszacowaniu jakości i mogą istnieć w pełni wielomianowe schematy obliczeń
b. mogą istnieć wielomianowe algorytmy aproksymacyjne o skończonym oszacowaniu jakości i mogą istnieć w pełni wielomianowe schematy obliczeń
c. nie mogą istnieć wielomianowe algorytmy aproksymacyjne o skończonym oszacowaniu jakości i nie mogą istnieć w pełni wielomianowe schematy obliczeń
d. mogą istnieć wielomianowe algorytmy aproksymacyjne o skończonym oszacowaniu jakości i nie mogą istnieć w pełni wielomianowe schematy obliczeń

42. Jeżeli P=NP to problem NP-trudny
- **a. to nie jest rozstrzygające**
- b. nie daje się rozwiązać w wielomianowym czasie
- c. daje się rozwiązać w wielomianowym czasie

43. To że jakiś problem optymalizacji kombinatorycznej jest matroidem pociąga za sobą, że:
- a. rozwiązanie puste nie musi być dopuszczalne i zawsze istnieje ciąg kroków od rozwiązania pustego do innego dopuszczalnego
- **b. rozwiązanie puste jest dopuszczalne i zawsze istnieje ciąg kroków od rozwiązania pustego do innego dopuszczalnego**
- c. rozwiązanie puste nie musi być dopuszczalne i nie musi istnieć ciąg kroków od rozwiązania pustego do innego dopuszczalnego

d. rozwiązanie puste jest dopuszczalne i nie musi istnieć ciąg kroków od rozwiązania pustego do innego dopuszczalnego

44. Między domknięciem przechodnim binarnej relacji endogennej, mnożeniem macierzy i wyznaczaniem odległości w grafie jest taki związek, że przez podmianę operatorów dodawania, maksimum, mnożenia, logicznych i/lub

a. algorytm wyznaczania odległości można zastąpić mnożeniem
b. algorytmu wyznaczania domknięcia przechodniego nie można zastąpić mnożeniem
c. algorytmu wyznaczania domknięcia przechodniego nie można zastąpić wyznaczaniem odległości
d. algorytm wyznaczania domknięcia przechodniego można zastąpić mnożeniem

45. W problemie wyznaczania maksymalnego przepływu w sieci rozwiązanie jest optymalne gdy

a. istnieje przekrój o pojemności większej niż wartość przepływu
b. istnieje przekrój o pojemności równej wartości przepływu
c. w wszystkich przekrojach pojemność przekroju jest mniejsza niż wartość przepływu
d. w wszystkich przekrojach pojemność przekroju jest większa niż wartość przepływu

46. Jeżeli jakiś algorytm aproksymacyjny dla jakiejś instancji konstruuje rozwiązanie o jakości 1.5*OPT, to bezwzględne oszacowanie jakości dla tego algorytmu jest

a. To niczego nie determinuje
b. Mniejsze lub równe 1.5
c. Równe 1.5
d. Większe lub równe 1.5

Bezwzględne oszacowanie odnosi się do najgorszego przypadku

47. W problemie wyznaczania przepływu o zadanej wartości F i minimalnym koszcie, przepływ jest optymalny gdy

a. nie istnieje cykl o ujemnym koszcie
b. istnieje cykl o ujemnym koszcie
c. istnieje ścieżka od źródła do ujścia o ujemnym koszcie
d. istnieje przekrój o pojemności równej wartości przepływu F
e. Nie istnieje cykl o ujemnym koszcie i wartość przepływu wynosi F
f. Istnieje cykl o ujemnym koszcie i wartość przepływu wynosi F
g. Wszystkie ścieżki od źródła do ujścia mają dodatnie koszty
h. Osiągniemy wartość przepływu równą F

Twierdzenie IIIc1 [Jewell, Busacker, Gowan]
Przepływ o wartości F posiada minimalny koszt wtedy i tylko wtedy, gdy w sieci nie występuje cykl powiększający przepływ o ujemnym koszcie.

Twierdzenie IIIc2 [JBG]
Powiększanie przepływu o minimalnym koszcie i wartości F o wartość ε wzdłuż ścieżki o minimalnym koszcie, powiększającej przepływ, prowadzi do konstrukcji przepływu o minimalnym koszcie i wartości F+ε.

48. Dla problemu plecakowego podano wielomianowy algorytmy aproksymacyjne A, B spełniające dla każdej instancji | warunki 1) algorytm A: |A(I)-OPT(I)|<=k, 2) algorytm B: B(I)/OPT(I)<=k, gdzie k>1 jest stałą, A(I), B(I) jakość rozwiązań aproksymacyjnych, OPT(I) skojarzenie wartość optymalna. Z tego wynika, że:

a. z 1) wynika, że P!=NP, z 2) wynika że P=NP
b. z 1) nic nie wynika, z 2) wynika że P=NP
c. z 1) **wynika, że P=NP, z 2) nic nie wynika**
d. z 1) wynika, że P=NP, z 2) wynika że P!I=NP

49. Skojarzenie w grafie, to

a. zbiór krawędzi które nie dają się pokolorować tym samym kolorem w problemie krawędziowego kolorowania grafu
b. zbiór krawędzi które dają się pokolorować tym samym kolorem w problemie krawędziowego kolorowania grafu
c. zbiór wierzchołków które dają się pokolorować tym samym kolorem w problemie wierzchołkowego kolorowania grafu
d. zbiór wierzchołków które nie dają się pokolorować tym samym kolorem w problemie wierzchołkowego kolorowania grafu

50. Jeżeli pewien problem optymalizacji jest silnie NP-zupełny w wersji decyzyjnej, to

a. P=NP pociąga istnienie w pełni wielomianowego schemat obliczeń (FPTAS)
b. istnienie w pełni wielomianowego schematu obliczeń (FPTAS) pociąga P=NP
c. istnienie w pełni wielomianowego schematu obliczeń pociąga P!=NP
d. P!=NP pociąga istnienie w pełni wielomianowy schemat obliczeń (FPTAS)

51. W problemie wyznaczania maksymalnego przepływu w sieci wartość przepływu liczona przy ujściu

a. jest mniejsza lub równa sumarycznej pojemności wszystkich łuków przecinanych przez każdy przekrój
b. jest inna w każdym przekroju
c. jest taka sama w każdym przekroju
d. jest zawsze równa sumarycznej pojemności wszystkich łuków przecinanych przez przekrój

52. Jeśli problem jest NP-łatwy

a. tak samo trudny jak cała klasa NP
b. to daje się rozwiązać w wielomianowym czasie
c. To jest nie trudniejszy niż klasa NP
d. to nie jest NP-trudny

53. Granice łatwości i trudności obliczeniowej problemu optymalizacji kombinatorycznej, można określić przez

a. zmianę założeń tego problemu i poszukiwanie przypadków łatwych
b. rozwiązania ciągłego problemu plecakowego
c. rozwiązywanie problemów podzielnego szeregowania zadań
d. zmianę założeń tego problemu i poszukiwanie przypadków łatwych i trudnych

54. W NP-zupełnym problemie podziału zbioru pyta się, czy zbiór liczb można podzielić na dwa podzbiory o równej sumie dobranych liczb. Z tego wynika, że w problemie pakowania liczb w minimalnej liczbie pudełek o stałej wysokości, nie istnieje wielomianowy algorytm aproksymacyjny

a. bezwzględnym oszacowaniu jakości równym 1.5
b. bezwzględnym oszacowaniu jakości równym 1
c. bezwzględnym oszacowaniu jakości równym 2
d. skończonym bezwzględnym oszacowaniu jakości

55. Problem domknięcia przechodniego binarnej relacji endogennej w ujęciu grafowym polega na tym żeby

a. dodać wszystkie łuki wynikające z symetrii relacji
b. dodać wszystkie łuki wynikające z symetrii i przechodniości relacji
c. dodać wszystkie możliwe łuki
d. dodać wszystkie łuki wynikające z przechodniości relacji

56. Porównując algorytmy dokładne i przybliżone dla problemów optymalizacji kombinatorycznej

a. algorytmy przybliżone mają wykładniczą złożoność, a dokładne nie
b. algorytmy dokładne dają gwarancje optymalności, a przybliżone nie
c. algorytmy dokładne mają wykładniczą złożoność, a przybliżone nie
d. algorytmy przybliżone dają gwarancje optymalności, a dokładne nie

57. Załóżmy, że w sieć warstwowa zbudowana przez algorytm Dinica do wyznaczania maksymalnego przepływu w sieci ma np. warstwy 1, 2, 3, wówczas:

a. łuki łączą warstwę 1 i F2 oraz 2 i 3 i mogą wystąpić łuki z warstwy 3 do 1
b. łuki łączą warstwę 1 i 2 oraz 2 i 3 i mogą wystąpić łuki wewnątrz każdej z warstw
c. łuki łączą warstwę 1 i 2 oraz 2 i 3 i mogą wystąpić łuki z warstwy 3 do 1 i wewnątrz każdej z warstw.
d. łuki łączą tylko warstwę 1 i 2 oraz 2 i 3

58. Problem kolorowania grafu jest

a. jest łatwy dla 2 kolorów i trudny dla 3 kolorów i dlatego nie istnieje algorytm aproksymacyjny o oszacowaniu lepszym niż 4/3

b. jest łatwy dla 3 kolorów i trudny dla 4 kolorów i dlatego nie istnieje algorytm aproksymacyjny o oszacowaniu lepszym niż 3

c. jest łatwy dla 2 kolorów i trudny dla 3 kolorów i dlatego nie istnieje algorytm aproksymacyjny o oszacowaniu lepszym niż 3

d. trudny obliczeniowo już od 2 kolorów i dlatego nie istnieje algorytm aproksymacyjny o oszacowaniu lepszym niż 1.5

e. jest łatwy dla 2 kolorów i trudny dla 3 kolorów i dlatego nie istnieje algorytm aproksymacyjny o oszacowaniu lepszym niż 3/2

f. jest łatwy dla 2 kolorów i trudny dla 3 kolorów i dlatego nie istnieje algorytm aproksymacyjny o oszacowaniu lepszym niż 4/3

g. jest łatwy dla 3 kolorów i trudny dla 2 kolorów i dlatego nie istnieje algorytm aproksymacyjny o oszacowaniu lepszym niż 3/2

h. jest łatwy dla 2 kolorów i trudny dla 3 kolorów i dlatego nie istnieje algorytm aproksymacyjny o oszacowaniu lepszym niż 3

59. Podanie bezwzględnego oszacowanie jakości algorytmu aproksymacyjnego

a. ogranicza od góry względną odległość rozwiązania od optimum

b. ogranicza od dołu względną odległość rozwiązania od optimum

c. ogranicza od góry bezwzględną odległość rozwiązania od optimum

d. ogranicza od dołu bezwzględną odległość rozwiązania od optimum

60. Jeżeli pewien problem jest silnie NP-zupełny i jego optymalizacyjny odpowiednik ma w pełni wielomianowy schemat obliczeń (FPTAS), to

a. P!=NP bo FPTAS jest wówczas algorytmem wielomianowym

b. P!=NP bo FPTAS jest wówczas algorytmem pseudowielomianowym

c. P=NP bo FPTAS jest wówczas algorytmem wielomianowym

d. P=NP bo FPTAS jest wówczas algorytmem pseudowielomianowym

Twierdzenie

Jeżeli dla trudnego problemu optymalizacji kombinatorycznej Π istnieje wielomian q taki, że

$$\forall I \in D_\Pi, OPT(I) < q(N(I), Max(I))$$

to istnienie w pełni wielomianowego aproksymacyjnego schematu obliczeń (FPTAS) dla Π pociągałoby za sobą istnienie pseudowielomianowego algorytmu optymalizacyjnego dla Π

Wniosek 1: Jeżeli Π spełnia warunki powyższego twierdzenia i jest s**NP**h, to istnienie FPTASa pociągałoby za sobą **P**=**NP**.

Wniosek 2: **P**$\neq$**NP**$\Rightarrow$ problemy s**NP**h nie mają FPTASów

61. Problem niepodzielnego szeregowania zadań na procesorach równoległych ma wielomianowe algorytmy aproksymacyjne

a. Nie ma o skończonej bezwzględnej odległości od optimum i ma o skończonej względnej odległości od optimum

b. Ma o skończonej bezwzględnej odległości od optimum i nie ma o skończonej względnej odległości od optimum

c. Nie ma o skończonej bezwzględnej odległości od optimum i nie ma o skończonej odległości od optimum

d. Ma o skończonej bezwzględnej odległości od optimum i ma o skończonej względnej odległości od optimum

Ma o skończonej względnej odległości od optimum => ma jakiś fajny algorytm typu 2-1/m

62. Problem komiwojażera ma wielomianowe algorytmy aproksymacyjne

a. Nie ma o skończonej bezwzględnej odległości od optimum i ma o skończonej względnej odległości od optimum

b. Ma o skończonej bezwzględnej odległości od optimum i nie ma o skończonej względnej odległości od optimum

c. Nie ma o skończonej bezwzględnej odległości od optimum i nie ma o skończonej odległości od optimum

d. Ma o skończonej bezwzględnej odległości od optimum i ma o skończonej względnej odległości od optimum

63. Jeżeli P=NP to problem NP-łatwy

a. Daje się rozwiązać w wielomianowym czasie

b. Nie daje się rozwiązać w wielomianowym czasie

c. To nie jest rozstrzygające

64. Jeżeli P jest różne od NP (P!=NP) to problem NP-łatwy

a. Daje się rozwiązać w wielomianowym czasie

b. To nie jest rozstrzygające

c. Nie daje się rozwiązać w wielomianowym czasie

65. W problemie wyznaczania maksymalnego przepływu w sieci z dolnymi i górnymi ograniczeniami na przepływ (po powrocie od sieci rozszerzonej do sieci pierwotnej), ścieżka powiększająca przepływ zawiera 1) łuki zwrócone od s do t i 2) wrócone od t do s, gdy

a. Łuki 1) mają przepływ większy od dolnego ograniczenia, łuki 2) mają przepływ mniejszy od pojemności

b. Łuki 1) mają przepływ większy od dolnego ograniczenia, łuki 2) mają przepływ zerowy

c. Łuki 1) mają przepływ równy pojemności, łuki 2) mają przepływ dodatni

d. Łuki 1) mają przepływ mniejszy od pojemności, łuki 2) mają przepływ większy od dolnego ograniczenia

66. Problem NP-trudny jest:
a. Z klasy NP
b. Co najwyżej tak trudny jak problemy NP-zupełne
c. Rozwiązywalny w wielomianowym czasie
d. Co najmniej tak trudny jak problemy NP-zupełne

67. Jeżeli P jest różne od NP (P!=NP) to
a. Żadne problemy NP-trudne nie dadzą się rozwiązać w wielomianowym czasie
b. Wszystkie problemy NP-równoważne dadzą się rozwiązać w wielomianowym czasie
c. Wszystkie problemy NP-trudne dadzą się rozwiązać w wielomianowym czasie
d. Wszystkie problemy NP-łatwe dadzą się rozwiązać w wielomianowym czasie

68. W problemie wyznaczania maksymalnego przepływu w sieci wartość przepływu liczona przy ujściu:
a. Jest mniejsza lub równa sumarycznej pojemności wszystkich łuków przecinanych przez każdy przekrój
b. Jest inna w każdym przekroju
c. Jest taka sama w każdym przekroju
d. Jest zawsze równa sumarycznej pojemności wszystkich łuków przecinanych przez przekrój

Jest jakaś pojemność ale niekoniecznie musisz nasycać <3

69. Jeżeli P jest różne od NP (P!=NP) to
a. Żadne problemy NP-trudne nie dadzą się rozwiązać w wielomianowym czasie
b. Wszystkie problemy NP-równoważne dadzą się rozwiązać w wielomianowym czasie
c. Wszystkie problemy NP-trudne dadzą się rozwiązać w wielomianowym czasie
d. Wszystkie problemy NP-łatwe dadzą się rozwiązać w wielomianowym czasie

70. W problemie wyboru algorytmu przestrzeń cech (feature space) jest
a. Jest stała między problemami optymalizacji kombinatorycznej
b. Ułatwia klasyfikację instancji
c. Ułatwia klasyfikację algorytmów
d. Niezbędna do rozwijania tego problemu

71. Problem równoważenia drzewa binarnego należy do klasy problemów
a. Przeszukiwania
b. Do klasy NP-zupełnych
c. Decyzyjnych
d. Do klasy NP-łatwych

72. Czy dla problemu plecakowego mogą istnieć wielomianowe algorytmy aproksymacyjne A,B spełniające dla każdej instancji I warunki 1) algorytm A: |A(I)-OPT(I)<=k, gdzie k>0 jest stałą, 2) algorytm B: OPT(I)/B(I) <= k?
a. 1 - tak, 2 - tak
b. 1 - tak, 2 - nie
c. 1 - nie, 2 - tak
d. 1 - nie, 2 - nie

73. Jeżeli takie oszacowania istnieją, to bezwzględne oszacowanie jakości algorytmu aproksymacyjnego jest
a. Większe lub równe asymptotycznemu oszacowaniu jakości
b. Mniejsze od asymptotycznego oszacowania jakości
c. Mniejsze lub równe asymptotycznemu oszacowaniu jakości
d. Większe od asymptotycznego oszacowania jakości

74. Problem plecakowy ma wielomianowe algorytmy aproksymacyjne
a. Ma o skończonej bezwzględnej odległości od optimum i nie ma skończonej względnej odległości od optimum
b. O skończonej bezwzględnej odległości od optimum i skończonej względnej odległości od optimum
c. Nie ma o skończonej bezwzględnej odległości od optimum i nie ma o skończonej względnej odległości od optimum
d. Nie ma o skończonej bezwzględnej odległości od optimum i ma o skończonej względnej odległości od optimum

75. Wyobraźmy sobie, że dla problemu komiwojażera (TSP), który jest silnie NP-zupełny w wersji decyzyjnej, podano wielomianowe algorytmy aproksymacyjne A,B spełniające dla każdej instancji I warunki 1) algorytm A który jest w pełni wielomianowym schematem obliczeń (FPTAS), 2) algorytm B: B(I)/OPT(I)<=k, gdzie 1<k? Wynika z tego, że:
a. Z 1) wynika, że P=NP, z 2) wynika że P!=NP
b. Z 1) wynika, że P=NP, z 2) nic nie wynika
c. Z 1) nic nie wynika, z 2) wynika że P=NP
d. Z 1) wynika, że P!=NP, z 2) wynika że P=NP

76. W problemie wyznaczania maksymalnego przepływu w sieci wartość przepływu jest
a. większa niż pojemność dowolnego przekroju
b. mniejsza niż pojemność dowolnego przekroju
c. większa lub równa pojemności dowolnego przekroju
d. mniejsza lub równa pojemności dowolnego przekroju

78. Algorytm symulowanego wyżarzania (simulated annealing) to

a. algorytm zachłanny
b. zaawansowana wersja hill climbera
c. algorytm populacyjny
d. uproszczona wersja hill climbera

79. Czy dla problemu plecakowego mogą istnieć wielomianowe algorytmy aproksymacyjne A,B spełniające dla każdej instancji I warunki 1) algorytm A: |A(I)-OPT(I)|<=K, gdzie k>0 jest stałą, 2) algorytm B: OPT(I)/B(I)<=k?

a. 1 - nie, 2 - tak
b. 1 - tak, 2 - tak
c. 1 - nie, 2 - nie
d. 1 - tak, 2 - nie

80. Algorytm zachłanny daje optymalne rozwiązanie problemu optymalizacji kombinatorycznej pod warunkiem, że

a. Rozwiązanie startowe jest puste
b. Gdy nie jest NP-zupełny
c. Ten problem jest z klasy P
d. Rozwiązanie startowe nie jest puste i jest wstępnie znane

82. W kolorowaniu krawędziowym, przy założeniu, że P!=NP, oszacowanie wartości indeksu chromatycznego jest:

a. Bliskie indeksu chromatycznego i nieobliczalne w wielomianowym czasie
b. Bliskie indeksu chromatycznego i obliczalne w wielomianowym czasie
c. ?
d. ?

Oszacowanie, a nie podanie dokładnej wartości

81. W kolorowaniu wierzchołkowym, przy założeniu P != NP, oszacowanie liczby chromatycznej jako mniejszej od lub równej stopniowi grafu plus 1 jest

a. Dowolnie odległe od liczby chromatycznej i nieobliczalne w wielomianowym czasie
b. Bardzo bliskie liczby chromatycznej i obliczalne w wielomianowym czasie
c. Bardzo bliskie liczby chromatycznej i nieobliczalne w wielomianowym czasie
d. Dowolnie odległe od liczby chromatycznej i obliczalne w wielomianowym czasie

Def. *Liczba chromatyczna* grafu G jest to najmniejsza liczba k taka, że istnieje pokolorowanie G za pomocą k kolorów i jest oznaczana symbolem $\chi(G)$.

• O ile $P \neq NP$ to dla problemu kolorowania, nie istnieje wielomianowy algorytm aproksymacyjny o oszacowaniu lepszym niż 2 [Garey, Johnson 1976].
O ile $P \neq NP$ to dla problemu kolorowania, wyznaczenie kolorowania $n^{\epsilon}\chi(G)$ kolorami jest NP-trudne, dla dowolnego $\epsilon > 0$.

82. Problem przydziału z N stanowiskami można rozwiązać przez sprowadzenie go do wyznaczania przepływu o wartości F=N i minimalnym koszcie, wykorzystując wyznaczanie najkrótszych ścieżek w grafie z uogólnionymi odległościami, ten algorytm ma złożoność O(F*N^3), gdzie O(N^3) to koszt wyznaczania odległości w grafie, ale złożoność tego algorytmu można

a. Ulepszyć stosując algorytm dla skierowanych grafów acyklicznych (DAG)
b. Ulepszyć stosując algorytm Forda-Bellmana
c. Ulepszyć w przeciętnym przypadku stosując algorytm Moore'a, Bellmana, d'Esopo, Pape'ego
d. Ulepszyć stosując algorytm Dijkstry

83. Graf GT trudny do kolorowania algorytmem A od dość trudnego GD do kolorowania algorytmem A różni się tym, że:

a. GT będzie pokolorowany nieoptymalnie tylko przy pewnej ustalonej numeracji wierzchołków, a GD będzie pokolorowany optymalnie przy pewnej określonej numeracji wierzchołków
b. GT będzie zawsze pokolorowany nieoptymalnie, a GD będzie pokolorowany nieoptymalnie tylko w losowych sytuacjach
c. GT będzie pokolorowany nieoptymalnie tylko przy pewnej ustalonej numeracji wierzchołków, a GD będzie pokolorowany nieoptymalnie w losowych sytuacjach
d. GT będzie zawsze pokolorowany nieoptymalnie, a GD będzie pokolorowany optymalnie przy pewnej określonej numeracji wierzchołków

• **Def.** Graph G is **hard to color** for algorithm A if any implementation of algorithm A colors G non-optimally.

• **Def.** Graph G is **slightly hard to color** for algorithm A if there is some implementation of algorithm A which colors G non-optimally.

84. W kolorowaniu wierzchołkowym, przy założeniu P != NP, oszacowanie liczby chromatycznej równe 4 dla grafów płaskich (planarnych) jest:

a. Dowolnie odległe od liczby chromatycznej i nieobliczalne w wielomianowym czasie
b. Dowolnie odległe od liczby chromatycznej i obliczalne w wielomianowym czasie
c. Bliskie liczby chromatycznej i obliczalne w wielomianowym czasie
d. Bliskie liczby chromatycznej i nieobliczalne w wielomianowym czasie

85. Graf trudny do kolorowania dla algorytmu random sequential

a. Nie może istnieć bo można odgadnąć optymalne kolorowanie
b. Może istnieć ale jeszcze tego grafu nie odkryto
c. Może istnieć bo można odgadnąć graf dość trudny
d. Nie może istnieć bo problem jest NP-trudny

86. W kolorowaniu wierzchołkowym, przy założeniu P!=NP, oszacowanie liczby chromatycznej jako większej od lub równej liczbie klikowej grafu jest:

a. Dowolnie odległe od liczby chromatycznej i nieobliczalne w wielomianowym czasie
b. Dowolnie odległe od liczby chromatycznej i obliczalne w wielomianowym czasie
c. Bardzo bliskie liczby chromatycznej i obliczalne w wielomianowym czasie
d. Bardzo bliskie liczby chromatycznej i nieobliczalne w wielomianowym czasie

87. W kolorowanie krawędziowym, przy założeniu P!=NP, oszacowanie wartości indeksu chromatycznego jest

a. Dowolnie odległe od indeksu chromatycznego i nieobliczalne w wielomianowym czasie
b. Dowolnie odległe od indeksu chromatycznego i obliczalne w wielomianowym czasie
c. Bliskie indeksu chromatycznego i nieobliczalne w wielomianowym czasie
d. Bliskie indeksu chromatycznego i obliczalne w wielomianowym czasie

88. W kolorowaniu wierzchołkowym, przy założeniu P!=NP, oszacowanie liczby chromatycznej równe 2 dla grafów dwudzielnych jest

a. Bliskie liczby chromatycznej i nieobliczalne w wielomianowym czasie
b. Dowolnie odległe od liczby chromatycznej i obliczalne w wielomianowym czasie
c. Dowolnie odległe od liczby chromatycznej i nieobliczalne w wielomianowym czasie
d. Bliskie liczby chromatycznej i obliczalne w wielomianowym czasie

89. W kolorowaniu wierzchołkowym, przy założeniu P!=NP, oszacowanie liczby chromatycznej równe stopniowi grafu (Delta) dla grafów dwudzielnych jest

a. Dowolnie odległe od liczby chromatycznej i nieobliczalne w wielomianowym czasie
b. Dowolnie odległe od liczby chromatycznej i obliczalne w wielomianowym czasie
c. Bliskie liczby chromatycznej i nieobliczalne w wielomianowym czasie
d. Bliskie liczby chromatycznej i obliczalne w wielomianowym czasie

90. W krawędziowym kolorowaniu grafu kolorowanie to funkcja

a. Określona na wierzchołkach
b. Liczbowa
c. Określona na parach krawędzi
d. Jej wartościami są krawędzie

91. W kolorowaniu wierzchołkowym, przy założeniu P!=NP, oszacowanie liczby chromatycznej jako większej od lub równej liczbie wierzchołków podzielonej przez liczbę niezależną grafu jest

a. Dowolnie odległe od liczby chromatycznej i obliczalne w wielomianowym czasie
b. Bardzo bliskie liczby chromatycznej i obliczalne w wielomianowym czasie
c. Bardzo bliskie liczby chromatycznej i nieobliczalne w wielomianowym czasie
d. Dowolnie odległe od liczby chromatycznej i nieobliczalne w wielomianowym czasie

92. W wierzchołkowym kolorowaniu grafu kolorowanie to funkcja

a. określona na krawędziach
b. określona na parach wierzchołków
c. jej wartościami są wierzchołki
d. liczbowa

93. Twierdzenie o nieistnieniu darmowych obiadów (no free lunch) dla problemów optymalizacji kombinatorycznej oznacza, że

a. Żadna metoda nie rozwiązuje wszystkich problemów optymalizacji kombinatorycznej
b. Problemy optymalizacji kombinatorycznej są za trudne, aby rozwiązać je efektywnie
c. Tylko w niektórych okolicznościach, tylko niektóre algorytmy są efektywne
d. We wszelkich okolicznościach, tylko niektóre algorytmy są efektywne

Jeśli algorytm jest dobry dla pewnych instancji, to dla innych będzie chujowy ~ Janek Zając informatyka, semestr III, studia stacjonarne, Poznań, nr. Indeksu 155948

94. W algorytmach przeszukiwania lokalnego (local search)

a. Ruchy muszą być lokalne
b. Sąsiedztwo musi być duże
c. Sąsiedztwo musi być wprost zdefiniowane aby można w nim było znaleźć rozwiązanie
d. Rozwiązanie musi być wprost zdefiniowane aby wykonywać z niego ruchy

95. Porównując algorytmy Multistart Local Search(MLS) i iterated local search (ILS)

a. MLS częściowo niszczy optimum lokalne, ILS wielokrotnie startuje od nowa
b. MLS wielokrotnie startuje od nowa, ILS wielokrotnie startuje od nowa
c. MLS częściowo niszczy optimum lokalne, ILS częściowo niszczy optimum lokalne
d. MLS wielokrotnie startuje od nowa , ILS częściowo niszczy optimum lokalne

a. MLS ma pamięć, ILS nie ma pamięci
b. MLS nie ma pamięci, ILS nie ma pamięci
c. <u>MLS nie ma pamięci, **ILS ma pamięć**</u>
d. MLS ma pamięć, **ILS ma pamięć**

96. Porównując metody przeszukiwania lokalnego i populacyjne:

a. populacyjne podążają jedną trajektorią rozwiązania
b. przeszukiwania lokalnego podążają jedną trajektorią rozwiązania
c. przeszukiwania lokalnego i populacyjne podążają jedną trajektorią rozwiązania
d. ani przeszukiwania lokalnego ani populacyjne nie podążają jedną trajektorią rozwiązania

97. W problemie wyboru algorytmu przestrzeń cech (feature space) jest:

a. jest funkcją od instancji do rozwiązań

b**. jest funkcją od instancji do algorytmów**

c. jest funkcją od instancji do abstrakcyjnej przestrzeni

d. niezbędna do rozwiązania tego problemu

98. To że jakiś problem optymalizacji kombinatorycznej daje się rozwiązać do optymalności algorytmem zachłannym pociąga za sobą, że

a. **rozwiązane puste jest dopuszczalne i zawsze istnieje ciąg kroków od rozwiązania pustego do innego dopuszczalnego**

b. rozwiązane puste nie musi być dopuszczalne i nie musi istnieć ciąg kroków od rozwiązania pustego do innego dopuszczalnego

c. rozwiązane puste nie musi być dopuszczalne i zawsze istnieje ciąg kroków od rozwiązania pustego do innego dopuszczalnego

d. rozwiązanie puste jest dopuszczalne i nie musi istnieć ciąg kroków od rozwiązania pustego do innego dopuszczalnego

99. Skojarzenie w grafie jest maksymalne, gdy

a. nie istnieją w nim ścieżki naprzemienne

b. żaden z pozostałych warunków nie jest wystarczający, aby skojarzenie było maksymalne

c. nie istnieją w nim ścieżki o nieparzystej długości

d. nie istnieją w nim ścieżki zaczynające i kończące się w wierzchołku wolnym

100. Porównując metody przeszukiwania lokalnego i populacyjne

a. przeszukiwania lokalnego potrzebują mniej pamięci niż populacyjne

b. przeszukiwania lokalnego i populacyjne potrzebują tyle są nieporównywalne ze względu na ilość używanej pamięci

c. przeszukiwania lokalnego i populacyjne potrzebują tyle samo pamięci

d. przeszukiwania lokalnego potrzebują więcej pamięci niż populacyjne

e. populacyjne podążają jedną trajektorią rozwiązania

f. przeszukiwania lokalnego podążają jedną trajektorią rozwiązania

g. przeszukiwania lokalnego i populacyjne podążają jedną trajektorią rozwiązania

h. ani przeszukiwania lokalnego ani populacyjne nie podążają jedną trajektorią rozwiązania

101. W problemie wyznaczania maksymalnego przepływu w sieci

a. przepływy łukowe mogą być nieograniczone

b. przepływy łukowe są ograniczone i nieujemne

c. przepływy łukowe są nieograniczone i nieujemne

d. przepływy łukowe mogą być ujemne

102. Algorytm GRASP to metoda która

a. jest wersją algorytmu przeszukiwania lokalnego

b. wprowadza losowość dołączania elementów do rozwiązania

c. błądzi losowo

d. jest jedno-przebiegowa

Greedy randomized adaptive search procedurę

103. W kolorowaniu wierzchołkowym, przy założeniu P!=NP, oszacowanie liczby chromatycznej jako mniejszej od lub równej stopniowi grafu Delta (o ile nie jest cyklem o nieparzystej długości i nie zawiera kliki o rozmiarze Delta+1) jest

a. dowolnie odległe od liczby chromatycznej i nieobliczalne w wielomianowym czasie

b. bardzo bliskie liczby chromatycznej i obliczalne w wielomianowym czasie

c. dowolnie odległe od liczby chromatycznej i obliczalne w wielomianowym czasie

d. bardzo bliskie liczby chromatycznej i nieobliczalne w wielomianowym czasie

104. Porównując metody przeszukiwania lokalnego i populacyjne

a. przeszukiwania lokalnego wymieniają cechy między rozwiązaniami

b. przeszukiwania lokalnego i populacyjne wymieniają cechy między rozwiązaniami

c**. populacyjne wymieniają cechy między rozwiązaniami**

d. ani przeszukiwania lokalnego ani populacyjne nie wymieniają cech między rozwiązaniami

Przydatne algorytmy (zachęcam do wstawiania):

Floyd - jak się w pętli k będzie printować “dist” to wszystkie iteracje dostaniemy

Sekcja plecakowa:

Python skrypt do tego problemu (ciągłego): https://pastebin.com/9qvjFyPd

Solver do innego 0-1 (jest tutaj również ciągły, ale daje niepoprawne wyniki): https://augustineaykara.github.io/Knapsack-Calculator/

Podaj rozwiązanie następującej instancji ciągłego problemu plecakowego. Wpisz 1/0 gdy element wchodzi/nie wchodzi do plecaka (zaznacz 1 także gdy wchodzi częściowo). Wyznacz wartość plecaka, przemnóż przez 1000 i podaj wynik w zaokrągleniu do liczb całkowitych:

- Wartości elementów. w=[2, 2, 4, 9, 8, 1, 6, 7, 6]. Rozmiary elementów: s=[4, 5, 5, 7, 5, 1, 5, 5, 4]. Rozmiar plecaka: 10.

wartości	2	2	4	9	8	1	6	7	6
rozmiar	4	5	5	7	5	1	5	5	4

- Wartości elementów. w=[10, 9, 8, 10, 5, 9, 1, 1, 4]. Rozmiary elementów: s=[3, 9, 6, 6, 6, 7, 6, 4, 7]. Rozmiar plecaka: 11.

- Wartości elementów. w=[8, 9, 3, 3, 3, 8, 7, 1, 1]. Rozmiary elementów: s=[6, 1, 4, 8, 10, 5, 8, 4, 1]. Rozmiar plecaka: 11.

- Wartości elementów. w=[4, 6, 6, 5, 1, 10, 4, 9, 9]. Rozmiary elementów: s=[6, 5, 6, 9, 2, 8, 2, 10, 5]. Rozmiar plecaka: 13.

- Wartości elementów: w=[5, 5, 10,5, 9, 1, 2, 5, 2]. Rozmiary elementów: s=[5, 6, 7, 10, 10, 4, 9, 7, 5]. Rozmiar plecaka: 13.

- Wartości elementów: w=[10, 9, 7, 8, 10, 2, 5, 10, 5]. Rozmiary elementów: s=[4, 4, 7, 6, 1, 6, 1, 3, 4]. Rozmiar plecaka: 15.

- Wartości elementów: w=[9, 1, 7, 9, 2, 3, 7, 1, 7]. Rozmiary elementów: s=[10, 7, 10, 1, 8, 8, 8, 3, 7]. Rozmiar plecaka: 11.

- Wartości elementów: w=[10, 5, 8, 8, 3, 10, 9, 7, 8]. Rozmiary elementów: s=[10, 10, 2, 9, 2,5, 4, 9, 7]. Rozmiar plecaka: 10.

- Wartości elementów: w=[2, 9, 3, 2, 10, 2, 6, 10]. Rozmiary elementów: s=[3, 10, 9, 4, 6, 9, 10, 2]. Rozmiar plecaka: 10.

- Wartości elementów: w=[1, 8, 6, 1, 3, 10, 2, 5, 7]. Rozmiary elementów: s=[6, 8, 5, 7, 9, 3, 3, 7, 4]. Rozmiar plecaka: 15.

- Wartości elementów: w=[4, 3, 5, 2, 10, 2, 10, 2, 7]. Rozmiary elementów: s=[4, 3, 5, 2, 10, 2, 10, 2, 7]. Rozmiar plecaka: 13.

- Wartości elementów: w=[10, 5, 2, 9, 9, 2, 10, 10, 6]. Rozmiary elementów: s=[2, 10, 8, 4, 5, 7, 8, 3, 1]. Rozmiar plecaka: 10.

- Wartości elementów: w=[4, 8, 9, 1, 6, 9, 5, 10, 10]. Rozmiary elementów: s=[5, 2, 9, 7, 2, 2, 10, 11, 5]. Rozmiar plecaka: 15.

- Wartości elementów: w=[8, 4, 9, 5, 2, 5, 2, 8, 3]. Rozmiary elementów: s=[5, 5, 8, 7, 1, 1, 8, 9, 3]. Rozmiar plecaka: 11 .