Question: A Boolean formula is satisfiable if there is an assignment of TR...

A Boolean formula is *satisfiable* if there is an assignment of TRUE's and FALSE's to the variables that makes the formula TRUE. A Boolean circuit with one output wire is *satisfiable* if there is an assignment of TRUE's and FALSE's (or 1's and 0's) to the inputs that makes the output wire TRUE (or 1).

Your manager calls you into the office with the following comment:

> It turns out that not only do we need to determine whether a Boolean formula is satisfiable, but, if so, we need to find a satisfying assignment. Furthermore, the inputs are more complicated than we expected: Boolean formulas can not only have variables, but they can also have TRUE's and FALSE's. So an input might, for example, be
>
> $$((A \wedge (\overline{B} \wedge \text{TRUE})) \vee (\overline{A} \wedge C)) \wedge ((B \wedge \overline{C}) \vee \text{FALSE})$$
>
> Once again we need your unique skills. You have two weeks to write a fast program to determine if a formula is satisfiable, and, if so, to find the satisfying assignment.

You find a program

`satisfiable(`H`)`

that allows TRUE's and FALSE's in the input, H, and returns YES or NO depending on whether the Boolean formula is satisfiable. It runs in time $\Theta(n^r)$, for some constant $r \geq 1$, where n is the number of variables.

You also find a program

`substitute(`H, X, V`)`

that substitutes the value V for variable X in formula H, where V is either TRUE or FALSE, and returns the new formula. It runs in linear time.

13. (a) Show how to use (the Boolean formula satisfiability program) `satisfiable` (and `substitute`) to efficiently find a satisfying assignment. Write the pseudo-code.
    (b) How fast is your algorithm? Justify.

Your manager calls you into the office, compliments you on a great job, and continues:

> It turns out that not only do we need to find a satisfying assignment, but the assignment must minimize the number of variables set to TRUE.

You find a progam

`satisfiable_num(`H, k`)`

that allows TRUE's and FALSE's in the input, H, and returns YES or NO depending on whether the Boolean formula is satisfiable with at most k variables set to TRUE. It runs in time $\Theta(n^r)$, for some constant $r \geq 1$, where n is the number of variables.

14. (a) Show how to use (the Boolean formula satisfiability program) `satisfiable_num` (and `substitute`) to find a satisfying assignment that minimizes the number of variables set to TRUE. Write the pseudo-code.
    HINT: eurottesebtsumselbairavynamwohenimretedtsrif.
    HINT for hint: sdrawkcabtidaer.
    (b) How fast is your algorithm? Justify.

Show transcribed image text

Expert Answer

Rashul Chutani answered this
1,049 answers

Was this answer helpful? 3 0

13. (a)

```
# We need to try all options(TRUE/FALSE) of all the Variables
# to find the correct arrangement.
permutations <- permutate(0,1,n) [Find all permutions O(2^n)]
for permutation in permutations:
for X in variables:

if(permutation[i] == 1)
substitute(H,X,true)
else
substitute(H,X,false)
if(satisfiable(H)) return permutation
increment i
```

Post a question
Answers from our experts for your tough homework questions
Enter question
Continue to post
18 questions remaining

Snap a photo from your phone to post a question
We'll send you a one-time download link
888-888-8888 Text me
By providing your phone number, you agree to receive a one-time automated text message with a link to get the app. Standard messaging rates may apply.

My Textbook Solutions
Linear Algebra... 5th Edition
The... 3rd Edition
Probability and... 8th Edition
View all solutions

Computer Science Chegg tutors who can help right now
Adivardhan M. Shri Ramdeobaba C... 12
Renato C. University of São Pa... 41
S. G. 34
Find me a tutor

```
# to find the correct arrangement.
permutations <- permutate(0,1,n)
for permutation in permutations:
    for X in variables:
        if(permutation[i] == 1)
            substitute(H,X,true)
        else
            substitute(H,X,false)
        if(satisfiable(H)) return permutation

   increment i
```

(b) Number of permutations(arrangements) = (2^n)

Inside for loop - n steps

substitute - n steps

satisfiable O(n^r)

Hence the total time complexity of the algorithm is **O(2^n * n^(r + 2))**

14. The 2nd Hint has to be read backwards and says ***'read it backwards'*** and the 1st hint says determine the min k for which H can be satisfiable!

Code:

for k in [0,n] :
if(satisfiable_num(H,k)):
optimal_k = k

permutations = permutation(0,1,n,k)
# We find permutations in which there are only k ones.
for permutation in permutations:
for X in variables:
if(permutation[i] == 1)
substitute(H,X,true)
else
substitute(H,X,false)
if(satisfiable(H)) return permutation
increment i

```
for k in [0,n] :
    if(satisfiable_num(H,k)):
        optimal_k = k

permutations = permutation(0,1,n,k)
    # We find permutations in which there are only k ones.
    for permutation in permutations:
        for X in variables:
            if(permutation[i] == 1)
                substitute(H,X,true)
            else
                substitute(H,X,false)
            if(satisfiable(H)) return permutation
           increment i
```

(b) Time Complexity: First we find the optimal-k in n*n^r time and then run our algorithm!

O(n*n^r + n! * n^(r + 2)) = O (n!*n(r+2))

Notice the reduction in complexity because of having an extra function **satisfiable_num**

P.S. You can comment below the answer in case of any doubts and I will be happy to answer!

All the best!

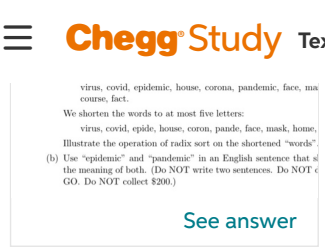
Please give a thumbs up if my answer could be of help to you!

View comments (1) ❯

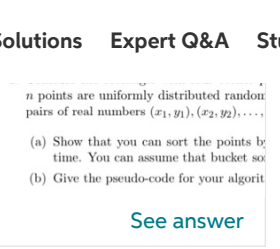
Up next for you in Computer Science

1. (a) Consider the following

2. Consider the rectangle with four corner points:


virus, covid, epidemic, house, corona, pandemic, face, ma
course, fact.
We shorten the words to at most five letters:
virus, covid, epide, house, coron, pande, face, mask, home,
Illustrate the operation of radix sort on the shortened "words".
(b) Use "epidemic" and "pandemic" in an English sentence that s
the meaning of both. (Do NOT write two sentences. Do NOT
GO. Do NOT collect $200.)

See answer


n points are uniformly distributed randon
pairs of real numbers $(x_1, y_1), (x_2, y_2), \ldots,$
(a) Show that you can sort the points b
time. You can assume that bucket so
(b) Give the pseudo-code for your algorit

See answer

for subjects you study

Questions viewed by other students

Q: A Boolean formula is satisfiable if there is an assignment of TRUE's and FALSE's to the variables that makes the formula TRUE. A Boolean circuit with one output wire is satisfiable if there is an assignment of TRUE's and FALSE's (or 1's and 0's) to the inputs that makes the output wire TRUE (or 1) 7. This question works with Boolean formulas in Conjuntive Normal Form (CNF). If you...

A: See answer 100% (1 rating)

Q: Your manager calls you into the office with the following comment: We are now moving into the business of Boolean formula satisfiability. Starting next month, every morning we will be receiving a large number of large Boolean formulas For each formula, we will need to determine if it is satisfiable. Note that we do not have to actually find the satisfying assignment; we just need a...

A: See answer

Show more

PayPal VERIFIED BBB OVER 6 MILLION TREES PLANTED

© 2003-2020 Chegg Inc. All rights reserved.