Puzzles

solution: reverse a string - word by word

problem: reverse "the house is blue", the answer should be "blue is house the". the words are reversed, but the letters are still in order (within the word).

solution: solving the initial problem of just reversing a string can either be a huge help or a frustrating hinderance. most likely the first attempt will be to solve it the same way, by swapping letters at the front of the string with letters at the back, and then adding some logic to keep the words in order. this attempt will lead to confusion pretty quickly.

for example, if we start by figuring out that "the" is 3 letters long and then try to put the "t" from "the" where the "l" from "blue" is, we encounter a problem. where do we put the "l" from "blue"? hmm... well we could have also figured out how long "blue" was and that would tell us where to put the "l" at... but the "e" from "blue" needs to go into the space after "the". argh. its getting quite confusing. in fact, i would be delighted to even see a solution to this problem using this attack method. i don't think its impossible, but i think it is so complex that it's not worth pursuing.

here's a hint. remember before when we just reversed "the house is blue"? what happened?

```
initial: the house is blue
reverse: eulb si esuoh eht
```

look at the result for a minute. notice anything? if you still don't see it, try this.

```
initial: the house is blue
reverse: eulb si esuoh eht
wanted : blue is house the
```

the solution can be attained by first reversing the string normally, and then just reversing each word

100 doors in a row aha:!

problem: you have 100 doors in a row that are all initially closed. you make 100 passes by the doors starting with the first door every time. the first time through you visit every door and toggle the door (if the door is closed, you open it, if its open, you close it). the second time you only visit every 2nd door (door #2, #4, #6). the third time, every 3rd door (door #3, #6, #9), etc, until you only visit the 100th door.

question: what state are the doors in after the last pass? which are open which are closed?

solution: doors

solution: 100 doors in a row

problem: you have 100 doors in a row that are all initially closed. you make 100 passes by the doors starting with the first door every time. the first time through you visit every door and toggle the door (if the door is closed, you open it, if its open, you close it). the second time you only visit every 2nd door (door #2, #4, #6). the third time, every 3rd door (door #3, #6, #9), etc, until you only visit the 100th door.

for example, after the first pass every door is open. on the second pass you only visit the even doors (2,4,6,8...) so now the even doors are closed and the odd ones are opened. the third time through you will close door 3 (opened from the first pass), open door 6 (closed from the second pass), etc..

question: what state are the doors in after the last pass? which are open which are closed?

solution: you can figure out that for any given door, say door #42, you will visit it for every divisor it has. so 42 has 1 & 42, 2 & 21, 3 & 14, 6 & 7. so on pass 1 i will open the door, pass 2 i will close it, pass 3 open, pass 6 close, pass 7 open, pass 14 close, pass 21 open, pass 42 close. for every pair of divisors the door will just end up back in its initial state. so you might think that every door will end up closed? well what about door #9. 9 has the divisors 1 & 9, 3 & 3. but 3 is repeated because 9 is a perfect square, so you will only visit door #9, on pass 1, 3, and 9... leaving it open at the end. only perfect square doors will be open at the end.

credit to joel wollman

red marbles, blue marbles aha:!

problem: you have two jars, 50 red marbles, 50 blue marbles. you need to place all the marbles into the jars such that when you blindly pick one marble out of one jar, you maximize the chances that it will be red. (when picking, you'll first randomly pick a jar, and then randomly pick a marble out of that jar) you can arrange the marbles however you like, but each marble must be in a jar.

solution: red marbles, blue marbles

solution: red marbles, blue marbles

problem: you have two jars, 50 red marbles, 50 blue marbles. you need to place all the marbles into the jars such that when you blindly pick one marble out of one jar, you maximize the chances that it will be red. (when picking, you'll first randomly pick a jar, and then randomly pick a marble out of that jar) you can arrange the marbles however you like, but each marble must be in a jar.

solution: chance! chance is easy if you know how to do the formula. we know that we have two choices to make. first we'll pick a jar, and each jar will have a 1/2 chance of being picked. then we'll pick a marble, and depending how we stack the marbles, we'll have a (# of red marbles in jar)/(# of total marbles in jar) chance of getting a red one.

for example, say we put all the red marbles into **jar A** and all the blue ones into **jar B**. then our chances for picking a red one are:

1/2 chance we pick **jar A** * 50/50 chance we pick a red marble
1/2 chance we pick **jar B** * 0/50 chance we pick a red marble

do the math and you get 1/2 chance for a red marble from **jar A** and a 0/2 chance for a red marble from **jar B**. add 'em up and you get the result = 1/2 chance for picking a red marble.

think about it for awhile and see if you can figure out the right combination. we had a 50/50 (guaranteed) chance in picking a red marble from **jar A**, but we didn't have to have 50 red marbles in there to guarantee those fantastic odds, did we? we could've just left 1 red marble in there and the odds are still 1/1. then we can take all those other marbles and throw them in **jar B** to help the odds out there.

let's look at those chances:

1/2 we pick **jar A** * 1/1 we pick a red marble
1/2 we pick **jar B** * 49/99 we pick a red marble

do the math and add them up to get 1/2 + 49/198 = 148/198, which is almost 3/4.

we can prove these are the best odds in a somewhat non-formal way as follows. our goal is to maximize the odds of picking a red marble. therefore we can subdivide this goal into maximizing the odds of picking a red marble in **jar A** and maximizing the odds of picking a red marble in **jar B**. if we do that, then we will have achieved our goal. it is true that by placing more red marbles into a jar we will increase the chances of picking a red marble. it is also true that by reducing the number of blue marbles in a jar we will increase the odds also. we've maximized the odds in **jar A** since 1/1 is the maximum odds by reducing the number of blue marbles to 0 (the minimum). we've also maximized the number of red marbles in **jar B**. if we added any more red marbles to **jar B** we would have to take them out of **jar A** which reduce the odds there to 0 (very bad). if we took any more blue ones out of **jar B** we would have to put them in **jar A** which reduce the odds there by 50% (very bad).

it wasn't really a good proof, but QED anyway :-P

bumblebee aha:!!!!

problem: two trains enter a tunnel 200 miles long (yeah, its a big tunnel) travelling at 100 mph at the same time from opposite directions. as soon as they enter the tunnel a supersonic bee flying at 1000 mph starts from one train and heads toward the other one. as soon as it reaches the other one it turns around and heads back toward the first, going back and forth between the trains until the trains collide in a fiery explosion in the middle of the tunnel (the bee survives). how far did the bee travel?

solution: bumblebee

solution: bumblebee

problem: two trains enter a tunnel 200 miles long (yeah, its a big tunnel) travelling at 100 mph at the same time from opposite directions. as soon as they enter the tunnel a supersonic bee flying at 1000 mph starts from one train and heads toward the other one. as soon as it reaches the other one it turns around and heads back toward the first, going back and forth between the trains until the trains collide in a fiery explosion in the middle of the tunnel (the bee survives). how far did the bee travel?

solution: this puzzle falls pretty high on my aha scale. my first inclination when i heard it was to think "ok, so i just need to sum up the distances that the bee travels..." but then you quickly realize that its a difficult (not impossible) summation which the interviewer could hardly expect you to answer (unless i guess if you are looking for a job as a quant). "there must be a trick" you say. eh, sort of i guess, enough to say that this question is a stupid interview question.

the tunnel is 200 miles long. the trains meet in the middle travelling at 100 mph, so it takes them an hour to reach the middle. the bee is travelling 1000 mph for an hour (since its flying the whole time the trains are racing toward one another) - so basically the bee goes 1000 miles.

there is no process to explain, so this question can't possibly teach you anything about the person. they either know it or they don't and if they already knew it before you asked, you're not going to be able to tell when they give you the answer. so don't ask this question. and if someone asks you this question, just tell them you've already heard it before.

int atoi(char* pStr) aha:!

problem: write the definition for this function *without using any built-in functions*. if pStr is null, return 0. if pStr contains non-numeric characters, either return 0 (ok) or return the number derived so far (better) (e.g. if its "123A", then return 123). assume all numbers are positive. plus or minus signs can be considered non-numeric characters. in order to solve this program, the programmer must understand the difference between the integer 0 and the character '0', and how converting '0' to an int, will not result in 0. in other words, they have to understand what ascii is all about.

solution: atoi

solution: atoi

string manipulation functions are great programming questions. they test whether the user can understand and translate into code simple algorithms. string functions test pointer arithmetic which usually shows a knowledgeable programmer. also there are usually multiple solutions, some more efficient than others. plus people use them all the time so they should understand how they work. my favorite is atoi and i start the problem like this:

```
int atoi( char* pStr )
```

write the definition for this function *without using any built-in functions*. if pStr is null, return 0. if pStr contains non-numeric characters, either return 0 (ok) or return the number derived so far (better) (e.g. if its "123A", then return 123). assume all numbers are positive. plus or minus signs can be considered non-numeric characters. in order to solve this program, the programmer must understand the difference between the integer 0 and the character '0', and how converting '0' to an int, will not result in 0. in other words, they have to understand what ascii is all about. if they are stuck solving this problem, just ask them first to write:

```
charToInt(char c)
```

if they can't do that then they basically missed half the problem. any moderately talented programmer who has a CS degree knows how to convert a char to an int. (note i said convert, not cast. `charToInt('9')` should return 9.)

when they start to solve the problem you will notice that they must make a choice in how they will process the string - from left to right or right to left. i will discuss both methods and the difficulties encountered in each.

"right to left" - this method starts at the right hand letter of the string and converts that character to an int. it then stores this value after promoting it to its correct "tens" place.

```
int atoi( char* pStr )
{
  int iRetVal = 0;
  int iTens = 1;

  if ( pStr )
  {
    char* pCur = pStr;
    while (*pCur)
      pCur++;

    pCur--;

    while ( pCur >= pStr && *pCur <= '9' && *pCur >= '0' )
    {
      iRetVal += ((*pCur - '0') * iTens);
      pCur--;
      iTens *= 10;
    }
  }
  return iRetVal;
}
```

"left to right" - this method keeps adding the number and multiplying the result by ten before continuing to the next number. e.g. if you had "6234" and you processed from left to right you'd have 6, then if you kept reading you'd multiply your result by 10 (6*10) to add a zero for where the next number would go. 60, and then you'd slide the 2 into the zero place you just made. 62. do it again, 620, slide the next number in, 623.

```
int atoi( char* pStr )
{
  int iRetVal = 0;

  if ( pStr )
  {
    while ( *pStr && *pStr <= '9' && *pStr >= '0' )
    {
      iRetVal = (iRetVal * 10) + (*pStr - '0');
      pStr++;
    }
  }
  return iRetVal;
}
```

i think the "left to right" method is a little bit cleaner, or maybe its just cooler. but both are "correct".

remember that debugging code on paper is somewhat hard. most programmers aren't used to studying code that much when you can just hit F-7, compile and see if the compiler barfs or not. if you notice an error, just ask them to step through a sample string drawing out what is happening with all the variables and the pointers in every step. they should find their mistake then and fix it (no points deducted).

daughter's ages aha:!!

two MIT math grads bump into each other at Fairway on the upper west side. they haven't seen each other in over 20 years.

the first grad says to the second: "how have you been?"
second: "great! i got married and i have three daughters now"
first: "really? how old are they?"
second: "well, the product of their ages is 72, and the sum of their ages is the same as the number on that building over there.."
first: "right, ok.. oh wait.. hmm, i still don't know"
second: "oh sorry, the oldest one just started to play the piano"
first: "wonderful! my oldest is the same age!"

problem: how old are the daughters?

solution: daughters

solution: daughter's ages

two MIT math grads bump into each other at Fairway on the upper west side. they haven't seen each other in over 20 years.

the first grad says to the second: "how have you been?"
second: "great! i got married and i have three daughters now"
first: "really? how old are they?"
second: "well, the product of their ages is 72, and the sum of their ages is the same as the number on that building over there.."
first: "right, ok.. oh wait.. hmm, i still don't know"
second: "oh sorry, the oldest one just started to play the piano"
first: "wonderful! my oldest is the same age!"

problem: how old are the daughters?

solution: start with what you know. you know there are 3 daughters whose ages multiply to 72. let's look at the possibilities...

Ages:	**Sum of ages**:
1 1 72	74
1 2 36	39
1 3 24	28
1 4 18	23
1 6 12	19
1 8 9	18
2 2 18	22
2 3 12	17

2 4 9	15
2 6 6	14
3 3 8	14
3 4 6	13

after looking at the building number the man still can't figure out what their ages are (we're assuming since he's an MIT math grad, he can factor 72 and add up the sums), so the building number must be 14, since that is the only sum that has more than one possibility.

finally the man discovers that there is an oldest daughter. that rules out the "2 6 6" possibility since the two oldest would be twins. therefore, the daughters ages must be "3 3 8".

(caveat: an astute reader pointed out that it **is** possible for two siblings to have the same age but not be twins, for instance one is born in january, and the next is conceived right away and delivered in october. next october both siblings will be one year old. if a candidate points this out, extra credit points to him/her.)

this question is pretty neat, although there is certainly a bit of an aha factor to it. the clues are given in such a way that you think you are missing information (the building number), but whats important isn't the building number, but the fact that the first man thought that it was enough information, but actually wasn't.

even if the candidate doesn't know the solution, they could come up with some interesting thoughts. if they just stare at you and shrug "i dunno" then thank them for their time and don't give them a fogcreek pen.

credit to david for reminding me of this one

palindromes aha:!!

problem: this year on October 2, 2001, the date in MMDDYYYY format will be a palindrome (same forwards as backwards).
10/02/2001
when was the last date that this occurred on? (see if you can do it in your head!)

solution: palindromes

thanks to tom, creator of atomize - the O(1) sorting algorithm

solution: palindrome

problem: this year on October 2, 2001, the date in MMDDYYYY format will be a palindrome (same forwards as backwards).
10/02/2001
when was the last date that this occurred on? (see if you can do it in your head!)

solution: we know the year has to be less than 2001 since we already have the palindrome for 10/02. it can't be any year in 1900 because that would result in a day of 91. same for 1800 down to 1400. it could be a year in 1300 because that would be the 31st day. so whats the latest year in 1300 that would make a month? at first i thought it would be 1321, since that would give us the 12th month, but we have to remember that we want the maximum **year** in the 1300 century with a valid month, which would actually be 1390, since 09/31 is a valid date.

but of course, a question like this wouldn't be complete without an *aha* factor. and of course, there are not 31 days in september, only 30. so we have to go back to august 08 which means the correct date would be **08/31/1380**.

palindromes also offer another great string question.
write a function that tests for palindromes
bool isPalindrome(char* pStr)

if you start a pointer at the beginning and the end of the string and keep comparing characters while moving the pointers closer together, you can test if the string is the same forwards and backwards. notice that the pointers only have to travel to the middle, not all the way to the other end (to reduce redundancy).

```
bool isPalindrome( char* pStr )
{
  if ( pStr == NULL )
   return false;

  char* pEnd = pStr;
  while ( *pEnd != '\0' )
    pEnd++;

  pEnd--;

  while(pEnd > pStr)
  {
    if ( *pEnd != *pStr )
      return false;

    pEnd--;
    pStr++;
  }

  return true;
}
```

thanks to tom for sending me this one! congrats on the wedding...

sum it up aha:!

problem: you are given a sequence of numbers from 1 to n-1 with one of the numbers repeating only once. (example: 1 2 3 3 4 5). how can you find the repeating number? what if i give you the constraint that you can't use a dynamic amount of memory (i.e. the amount of memory you use can't be related to n)? what if there are two repeating numbers (and the same memory constraint?)

solution: sum it up

solution: sum it up'

problem: you are given a sequence of numbers from 1 to n-1 with one of the numbers repeating once. (example: 1 2 3 3 4 5) how can you find the repeating number?

solution:

as a programmer, my first answer to this problem would be make a bit vector of size n, and every time you see the number, set its correspond index bit to 1. if the bit is already set, then that's the repeater. since there were no constraints in the question, this is an ok answer. its good because it makes sense if you draw it for someone, whether they are a programmer, mathemetician, or just your grandpa. its not the most efficient answer though.

now, if i add the constraint that you can only use a fixed amount of memory (i.e. not determined by n) and it must run in O(n) time... how do we solve it. adding all the numbers up from 1 to n-1 would give us a distinct sum. subtracting the total sum of all the numbers from the sum of n to n-1 (which is (n)(n-1)/2) would give us the secret extra number.

what if you can only use a fixed amount of memory, and **two** of the numbers are repeated? we know that the numbers have a distinct sum, and the difference would be equal to the sum of our unknowns
c = a + b
where c is the sum and *a* and *b* are the unknowns - c is a constant
if we had another similar formula we could solve the two unknown equations. my first thought was that the numbers would have a distinct product - (n-1)!
if we divide the total product by the (n-1)! product, we would get another equation
c2 = ab
we could then solve the two equations to get them into quadratic formula notation
0 = ax^2 + bx + c
and solve for the two values of x. this answer is correct but factorial grows really fast.

some sort of sum would be better. the sum of the squares from n-1 to 1 would work. that would yield a function of the form
c2 = a^2 + b^2
which could also be solved by using the quadratic equation.

i think its fine to remind someone of the quadratic equation... (maybe only because i myself had to look it up to solve the problem) i mean really though, the last time i used it was probably in 10th grade. as long as they get the idea that given two unknowns and two equations you can solve for the unknowns - thats the point.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

pirates aha:!

five pirates have 100 gold coins. they have to divide up the loot. in order of seniority (suppose pirate 5 is most senior, pirate 1 is least senior), the most senior pirate proposes a distribution of the loot. they vote and if at least 50% accept the proposal, the loot is divided as proposed. otherwise the most senior pirate is executed, and they start over again with the next senior pirate. what solution does the most senior pirate propose? assume they are very intelligent and extremely greedy (and that they would prefer not to die).

solution: pirates

solution: pirates

five pirates have 100 gold coins. they have to divide up the loot. in order of seniority (suppose pirate 5 is most senior, pirate 1 is least senior), the most senior pirate proposes a distribution of the loot. they vote and if at least 50% accept the proposal, the loot is divided as proposed. otherwise the most senior pirate is executed, and they start over again with the next senior pirate. what solution does the most senior pirate propose? assume they are very intelligent and extremely greedy (and that they would prefer not to die).

(to be clear on what 50% means, 3 pirates must vote for the proposal when there are 5 for it to pass. 2 if there are 4. 2 if there are 3. etc...)

solution: most of the time i get people who give answers like "the most senior pirate takes half and divides the rest up among the least senior pirates." um, you missed the whole point to begin with. sorry.

any answer without a specific logic behind it is invalid. if i ask you why pirate 5 gave x coins to pirate 1, please don't say "because he's nice".

now for the real solution. pirate 5 being the most senior knows that he needs to get 2 other people to vote for his solution in order for him not to be executed. so who can he get to vote for him, and why would they choose to vote for him? if you start thinking that pirate 4 will never vote for him, because he would rather have 5 die and then be in charge and take it all for himself, you are on the right track. but it gets more complicated.

lets consider if there were only 1 pirate. obviously he would take it all for himself and no one would complain.

if there were 2 pirates, pirate 2 being the most senior, he would just vote for himself and that would be 50% of the vote, so he's obviously going to keep all the money for himself.

if there were 3 pirates, pirate 3 has to convince at least one other person to join in his plan. so who can he convince and how? here is the leap that needs to be made to solve this problem. **pirate 3 realizes that if his plan is not adopted he will be executed and they will be left with 2 pirates**. he already knows what happens when there are 2 pirates as we just figured out. pirate 2 takes all the money himself and gives nothing to pirate 1. so pirate 3 proposes that he will take 99 gold coins and give 1 coin to pirate 1. pirate 1 says, well, 1 is better than none, and since i know if i don't vote for pirate 3, i get nothing, i should vote for this plan.

now we know what happens when there are 3 pirates. so what happens with 4? well pirate 4 has to convince 1 other person to join in his plan. he knows if he walks the plank then pirate 3 will get 99 coins and pirate 1 will get 1 coin. pirate 4 could propose giving pirate 1 two coins, and surely pirate 1 would vote for him, since 2 is better than 1. but as greedy as he is, pirate 4 would rather not part with 2 whole coins. he realizes that if he gets executed, then pirate 3's scenario happens and pirate 2 gets the shaft in that scenario (he gets zero coins). so pirate 4 proposes that he will give 1 coin to pirate 2, and pirate 2 seeing that 1 is better than 0 will obviously vote for this plan.

a common objection is that pirate 2 is not guaranteed to vote for this plan since he might hope for the case when there are only 2 pirates and then he gets all the booty. but that is why i said that the pirates are extremely intelligent. pirate 2 realizes that pirate 3 is smart enough to make the optimal proposal, so he realizes that there will never be 2 pirates left, because 3 doesn't want to die and we just showed that 3 has a winning proposal.

so lets sum up at this point

Pirate	1	2	3	4	5
5.	?	?	?	?	?
4.	0	1	0	99	-
3.	1	0	99	-	-
2.	0	100	-	-	-
1.	100				

once you see the pattern it becomes very clear. you have to realize that when a pirate's plan does not succeed then that means you are in the same situation with one less pirate.

1. pirate 1 needs 0 other people to vote for him. so he votes for himself and takes all the money. 2. pirate 2 needs 0 other people to vote for him. so he votes for

himself and takes all the money. pirate 1 gets 0. 3. pirate 3 needs 1 other person to vote for him. he gives 1 coin to pirate 1 for his vote - if we are reduced to 2 pirates, pirate 1 gets 0 so pirate 1 knows 1 is better than none. pirate 3 takes 99. pirate 2 gets 0. 4. pirate 4 needs 1 other person to vote for him. he gives 1 coin to pirate 2 - if we reduce to 3 pirates, pirate 2 gets 0 so pirate 2 knows 1 is better than none. pirate 4 takes 99. pirate 3 gets 0. pirate 1 gets 0. 5. pirate 5 needs 2 other people to vote for him. its clear now that the 2 people he needs to convince are the 2 who get shafted in the 4 pirate scenario - pirate 3 and pirate 1. so he can give them each 1 coin (which is better than 0 - what they would get otherwise) and keep 98 for himself.

Pirate	1	2	3	4	5
5.	1	0	1	0	98

what happens if there are 15 pirates? pirate 15 needs 7 other people to vote for him, so he recruits pirates 13,11,9,7,5,3, and 1 with 1 coin each and keeps 93 coins himself. those pirates will all vote for him because they know that they get 0 coins if he dies and pirate 14 is in charge.

hope you enjoyed this one. its my favorite interview question of all. it really allows the candidate to ask a lot of interesting questions and its really amazing when they reach the solution all by themselves (as all fogcreek employees have done so far).

fogcreek programmers aha:!

100 fogcreek programmers are lined up in a row by an assassin. the assassin puts red and blue hats on them. they can't see their own hats, but they can see the hats of the people in front of them. the assassin starts in the back and says "what color is your hat?" the fogcreek programmer can only answer "red" or "blue." the programmer is killed if he gives the wrong answer; then the assassin moves on to the next programmer. the programmers in front get to hear the answers of the programmers behind them, but not whether they live or die. they can consult and agree on a strategy before being lined up, but after being lined up and having the hats put on, they can't communicate in any way other than those already specified. what strategy should they choose to maximize the number of programmers who are guaranteed to be saved?

solution: fogcreek programmers

solution: fog creek programmers

problem: 100 fogcreek programmers are lined up in a row by an assassin. the assassin puts red and blue hats on them. they can't see their own hats, but they can see the hats of the people in front of them. the assassin starts in the back and says "what color is your hat?" the fogcreek programmer can only answer "red" or "blue." the programmer is killed if he gives the wrong answer; then the assassin moves on to the next programmer. the programmers in front get to hear the answers of the programmers behind them, but not whether they live or die. they can consult and agree on a strategy before being lined up, but after being lined up and having the hats put on, they can't communicate in any way other than those already specified. what strategy should they choose to maximize the number of programmers who are guaranteed to be saved?

this is a very difficult problem to solve during an interview (especially if you've already taxed the candidate's brain). look for obvious solutions first, and the reasoning behind them and then try to lead them to the ultimate solution.

a logical answer could be all the programmers would just say "red" and that way about half of them would survive on average, assuming the hats were distributed randomly.

this is a good start and should naturally lead to having every other programmer say the color of the hat in front of them. the first programmer would say the color of the hat in front of him, then the next programmer would just say that color that was just said. so we can guarantee that half survive - the even numbered programmers (since the person behind them told them the answer). and potentially if the hats were distributed randomly some of the programmers would get lucky and the hat in front of them would be the same color as their own. so this strategy should save more than half, and on average 75% of them would live.

at this point, if the solution is not clear, the candidate may give answers like, "they could agree that if they said their hat color in a soft voice, it means the hat in front of them is the same color, and if they say it in a loud voice, it means the hat in front is a different color". this is definitely good and on the correct track. another option is they could say "reeeeeeeeeeed" for x number of seconds, where x represented the distribution of hats where a hat was a bit in a binary number, (red = 1, blue = 0). another interesting answer. there are many others like these that "bend" the rules and come to a solution.

but the real solution acknowledges that the programmers can only say "red" or "blue" and cannot alter their voice in such a convincing way as to signal any information other than the word they said. a good way to get this point across, is simply to change the problem slightly by saying "the assassin gets to hear their plan before she puts the hats on, and so will try to thwart the plan however she can."

so if they decide to all say "red", she'll put blue hats on all of them. if they decide to all say the color of the hat in front of them, she'll alternate the hats on every head, guaranteeing half will die. even with the assassin hearing their plan, there is still a way to save almost everyone.

we know that the first person is never going to have any information about the color of their hat, so they cannot be guaranteed to survive. but, i'll give you a hint to the solution: i can save every other person for sure.

solution: they agree that if the number of red hats that the back person can see is even, that programmer will say "red". if they add up to an odd number, they will say "blue". this way number 99 can look ahead and count the red hats. if they add up to an even number and number 100 said "red", then 99 must be wearing a blue hat. if they add up to an even number and number 100 said "blue", signalling an odd number of red hats, number 99 must also be wearing a red hat. number 98 knows that 99 said the correct hat, and so uses that information along with the 97 hats in front to figure out what color hat is on 98's head.

sample:

```
100  99  98  97  96  95  94 ... facing ->
 R    B   B   R   B   R   B ... -> 45 R and 48 B
```

this shows #100 wearing a red hat, 99 a blue, 98 a blue, 97 a red, 96 a blue, 95 a red, 94 a blue and 45 red hats - 48 blue hats on the people in front of them.
100 counts up the red hats: 47 total. so 100 says "blue". the assassin kills 100. 99 counts up the red hats in front: 47. 100 said blue, so 100 saw an odd number. 99 sees an odd number, so 99 says "blue" and lives. 98 had counted 47 red hats, and 99 didn't say "red" so thats still the total. 98 says "blue". 97 counts up and finds 46 red hats. 99 and 98 didn't say "red", so his count is missing a red hat (its on his head, he realizes). he says "red". 96 heard the "red" and now knows that there are an even number of "red" hats in front of 95. 96 sees 46, so he knows he has a "blue" hat. etc...
even if the assassin knows the plan, she can't thwart it. she hears the plan, but she still has to put the hats on their heads. the plan doesn't rely on any ordering of the hats, so the worst the assassin can do is to make sure #100 gets killed and thats the worst damage she can do.

bad king aha:!

a bad king has a cellar of 1000 bottles of delightful and very expensive wine. a neighbouring queen plots to kill the bad king and sends a servant to poison the wine. (un)fortunately the bad king's guards catch the servant after he has only poisoned one bottle. alas, the guards don't know which bottle but know that the poison is so strong that even if diluted 1,000,000 times it would still kill the king. furthermore, it takes one month to have an effect. the bad king decides he will get some of the prisoners in his vast dungeons to drink the wine. being a clever bad king he knows he needs to murder no more than 10 prisoners - believing he can fob off such a low death rate - and will still be able to drink the rest of the wine at his anniversary party in 5 weeks time.

explain how....

thanks to Duncan Smeed for tipping me off to this question.

solution: bad king

solution: bad king

problem: a bad king has a cellar of 1000 bottles of delightful and very expensive wine. a neighbouring queen plots to kill the bad king and sends a servant to poison the wine. (un)fortunately the bad king's guards catch the servant after he has only poisoned one bottle. alas, the guards don't know which bottle but know that the poison is so strong that even if diluted 1,000,000 times it would still kill the king. furthermore, it takes one month to have an effect. the bad king decides he will get some of the prisoners in his vast dungeons to drink the wine. being a clever bad king he knows he needs to murder no more than 10 prisoners - believing he can fob off such a low death rate - and will still be able to drink the rest of the wine at his anniversary party in 5 weeks time.

explain how...

solution: i'll give you a hint. 1000 is less than 1024. if there were 1024 or more bottles of wine it would take more than 10 prisoners.

number the bottles 1 to 1000, and write the number in binary format.

bottle 1   = 0000000001
bottle 250 = 0011111010
bottle 1000 = 1111101000

now take your prisoner's 1 through 10 and let prisoner 1 take a sip from every bottle that has a 1 in its least significant bit. let prisoner 10 take a sip from every bottle with a 1 in its most significant bit. etc.

prisoner	10	9	8	7	6	5	4	3	2	1
bottle 924	1	1	1	0	0	1	1	1	0	0

for instance, bottle #924 would be sipped by 10,9,8,5,4 and 3. that way if bottle #924 was the poisoned one, only those prisoners would die.

after four weeks, line the prisoners up in their bit order and read each living prisoner as a 0 bit and each dead prisoner as a 1 bit. the number that you get is the bottle of wine that was poisoned.

additional question: to increase your chance of living, which prisoner would you want to be?

if there were 1023 bottles, it wouldn't matter since everyone would have to take 512 sips. but there are 23 bottles less, so the people whose bits would have been on from 1001 to 1023 won't have to take a sip. 1001 is [11111 01001] in binary and 1023 is [11111 11111]. the most five significant bits are the most interesting because they would always be on from 1001 to 1023, so all those people are missing out on 23 bottles of wine that they otherwise would have had

to drink. so in order to increase your chance of living, you'd probably want to be prisoner 6 to 10. (but depending on how the king determines who is least significant and who is most significant you could get shafted.)
note that if the king was really trying to kill the least number of prisoners, he should have let 999 prisoners each take a sip from their respective bottle numerically (if he had that many prisoners at his disposal). that way only one prisoner would die, and there's a chance of 1/1000 that no one would die, but then the puzzle isn't very fun.

jelly beans aha:!

you have three jars that are all mislabeled. one contains peanut butter jelly beans, another grape jelly jelly beans, and the third has a mix of both (not necessarily a 50/50 mix, could be a 1/99 mix or a 399/22 mix). how many jelly beans would you have to pull out, and out of which jars, to find out how to fix the labels on the jars?

```
|     |       |     |         |     |
|jar 1|       |jar 2|         |jar 3|
|     |       |     |         |     |
=======     =======       =======
  p.b.         grape         p.b./grape
```

thanks to joel wollman

solution: jelly beans

solution: jellybeans

you have three jars that are all mislabeled. one contains peanut butter jelly beans, another grape jelly jelly beans, and the third has a mix of both (not necessarily a 50/50 mix, could be a 1/99 mix or a 399/22 mix). how many jelly beans would you have to pull out, and out of which jars, to find out how to fix the labels on the jars?

```
|     |       |     |         |     |
|jar 1|       |jar 2|         |jar 3|
|     |       |     |         |     |
=======     =======       =======
  p.b.         grape         p.b./grape
```

solution: 1 jelly bean from the p.b./grape jar will do the trick.
the trick here is to realize that every jar is mislabeled. therefore you know that the peanut butter jelly bean jar is not the penut butter jelly bean jar, and the same goes for the rest.
you also need to realize that it is the jar labeled p.b./grape, labelled as the mix jar, that is your best hope. if you choose a jelly bean out of there, then you will

know whether that jar is peanut butter or grape jelly jelly beans. it can't be the mix jar because i already said that every jar is mislabeled.
once you know that jar 3 is either peanut butter, or grape jelly, then you know the other jars also. if it is peanut butter, then jar 2 must be mixed because it can't be grape (as its labelled) and it can't be peanut butter (that's jar 3). hence jar 1 is grape.
if jar 3 is grape, then you know jar 1 must be the mix because it can't be p.b. (as its labelled) and it can't be grape (that's jar 3). hence jar 2 is peanut butter.
if you pick jelly beans from jar 1 or jar 2, then you would have to pick out **all** of the jelly beans before you knew what that jar was. this is because jar 1 and 2 could be the mix, so in order to disprove that they were the mix, you would have to pull out **every** jelly bean just to make sure (since there could just be one bean of the opposite flavor in there).

bridge aha:!!

problem: this one is a classic that many of you have probably already heard, but all the more reason why it should definitely be included here. four people are on this side of the bridge. the bridge will be destroyed by a bomb in 17 minutes. everyone has to get across before that. problem is that it's dark and so you can't cross the bridge without a flashlight, and they only have one flashlight. plus the bridge is only big enough for two people to cross at once. the four people walk at different speeds: one fella is so fast it only takes him 1 minute to cross the bridge, another 2 minutes, a third 5 minutes, the last it takes 10 minutes to cross the bridge. when two people cross the bridge together (sharing the flashlight), they both walk at the slower person's pace. can they all get across before the bridge blows up?

person A: 1 minute
person B: 2 minutes
person C: 5 minutes
person D:10 minutes

solution: bridge

solution: bridge

problem: four people are on this side of the bridge. the bridge will be destroyed by a bomb in 17 minutes. everyone has to get across before that. problem is that it's dark and so you can't cross the bridge without a flashlight, and they only have one flashlight. plus the bridge is only big enough for two people to cross at once. the four people walk at different speeds: one fella is so fast it only takes him 1 minute to cross the bridge, another 2 minutes, a third 5 minutes, the last it takes 10 minutes to cross the bridge. when two people cross the bridge together (sharing the flashlight), they both walk at the slower person's pace. can they all get across before the bridge blows up?

person A: 1 minute
person B: 2 minutes
person C: 5 minutes
person D:10 minutes

solution: of course its possible, otherwise it wouldn't be a very interesting question. the only trick is in realizing that you want to get the two slowest people across together, because otherwise you are wasting too much time. but then once you get them across, how do you not make one of them walk back with the flashlight? well, you just have one of the fast people already there waiting to sprint the flashlight back across.

1. A & B cross. total time: 2 minutes.

```
 C |==========================| A
 D |                          | B
   |==========================| flashlight
```

2. B comes back. total time: 4 minutes.

```
 C |==========================| A
 D |                          |
 B |==========================|
flashlight
```

3. C & D cross. total time: 14 minutes.

```
 B |==========================| A
   |                          | C
   |==========================| D
                        flashlight
```

4. A comes back. total time: 15 minutes.

```
 A |==========================| C
 B |                          | D
   |==========================|
flashlight
```

5. A & B cross. total time: 17 minutes.

```
   |=========    =============| A
   |       KABOOM!            | B
   |=========    =============| C D
                        flashlight
```

another solution which is valid is to have A bring the flashlight back in step 2. it only changes the solution slightly. this is supposed to be a "classic" microsoft interview question but it seems absurdly easy to be a good interview question (especially coupled with the fact that everyone has probably heard it already).

card trick without the trick aha:!!

this is a card trick without the trick. there is no sleight of hand, no tricks up my sleeve, no magic whatsoever. it's all done with logic, yet it will amaze most people.

joel and i are working together as a team to do the trick. babak will be the culprit.

i ask babak to pick 5 cards out of a deck. he can pick any five cards, he can shuffle the deck 7 times, it really doesn't matter. he honestly picks out 5 cards that i cannot see. he hands the five cards to me (joel can't see any of this). i look at the cards and i pick 1 card out and give it back to babak. i then arrange the other four cards in a special way, and give those 4 cards all face down, and in a neat pile, to joel. joel looks at the 4 cards i gave him, and says out loud which card babak is holding (suit and number).

i did not convey any information to joel other than the way i ordered the 4 cards, (all face down, aligned in line with one another) so how did i encode babak's card using that method?

hint 1: card trick without the trick
hint 2: card trick without the trick

solution: card trick without the trick

solution: card trick without a trick

this is a card trick without the trick. there is no sleight of hand, no tricks up my sleeve, no magic whatsoever. it's all done with logic, yet it will amaze most people.

joel and i are working together as a team to do the trick. babak will be the culprit.

i ask babak to pick 5 cards out of a deck. he can pick any five cards, he can shuffle the deck 7 times, it really doesn't matter. he honestly picks out 5 cards that i cannot see. he hands the five cards to me (joel can't see any of this). i look at the cards and i pick 1 card out and give it back to babak. i then arrange the other four cards in a special way, and give those 4 cards all face down, and in a neat pile, to joel. joel looks at the 4 cards i gave him, and says out loud which card babak is holding (suit and number).

i did not convey any information to joel other than the way i ordered the 4 cards, (all face down, aligned in line with one another) so how did i encode babak's card using that method?

hint 1: card trick without the trick
hint 2: card trick without the trick

solution: hint 1 talks about how there are guaranteed to be at least 2 cards with the same suit in a series of five cards. i use this to signal to joel the suit of the card. i have to make a decision though between which of the two cards to give back to babak. which one do i pick?

hint 2 talks about how if i use one of the cards to signal the suit, that only leaves me with 3 cards. i can only arrange 3 cards in six permutations. how do i signal a number between 2 and 13 using only six permutations?

simple really. consider that the series of cards is a cycle.

2 3 4 5 6 7 8 9 10 J Q K A 2 3 4 5 6 7 8 9 10 J Q K A 2 ... etc

i take the two cards of the same suit (if there are more than two of the same suit, then i choose any two, it won't affect the solution) and i pass babak the one that is further to the right in the cycle. if i do it correctly, the one further to the right will be no more than 6 places away than the one to the left. (it just depends on how you think about the cycle).

for example. say we had the 2 and the 9 of hearts. if you look at it as the 2 to the left and the 9 to the right, it is seven cards away. but if you look as the 9 to the left and the 2 to the right in the cycle, the 2 is only 6 cards away.

then i just come up with a system of the permutations that signals how many places to the right to count to find the correct card

one such system could be:

sort the 3 non-same suit cards first by number then by suit (diamonds, spades, hearts, clubs) to give a unique sort for every card.

label the cards as A, B, and C.

place the cards in the following order to signal how many places away babak's card is:

A B C - 1
A C B - 2
B A C - 3
B C A - 4
C A B - 5
C B A - 6

then put the suit card on top and give the pile to joel.

example:

babak picks out the following five cards:

3C 7D 10S QH AD

the 7 of diamonds (7D) and the ace of diamonds (AD) are the two suit cards. i recognize that AD is going to be the left card and 7D the right (because the other way they are 7 spaces apart). i give babak back the 7D. i sort the other cards, 3C 10S QH. i have to signal 6 places to the right (7D - AD = 6). so i place them in C B A format: QH 10S 3C. i then place the AD on top and pass to joel.

joel pulls off the first card and sees that it is the AD. (he knows babak's card is a diamond). he looks at the next 3 cards and realizes their ordering is C B A.

signalling 6, he counts 6 spots from the ace. (2 3 4 5 6 7). then he says "7 of diamonds" and the crowd is amazed.
another **sneakier card trick** (with the trick) works as follows.
babak picks one card out of the deck and shows it to me. he doesn't tell joel and i don't speak. after seeing the card, i sneakily use my one hand to signal both the number and suit of the card. i use my four fingers (not the thumb) as bits signalling a binary number. if they are extended it means 1, if i don't extend them it means 0. the pinky is the least significant. so to signal 7 i would extend all fingers except my index finger. the suit is signalled by four positions of the thumb (hidden, adjacent to the hand, slightly ajar, fully extended). i can use this method to tell joel the card without saying anything (and making any noticeable motions). with some practice it can really amaze people (at least it worked with my family).

pirates revisited aha:!

a slightly different version of the original pirates problem (read that one first to get all the rules). 6 pirates, only one gold coin. as before, the pirates are super-smart, and they value, in this order: (i) their lives, (ii) getting money, (iii) seeing other pirates die. so if given the choice between two outcomes, in which they get the same amount of money, they'd choose the outcome where they get to see more of the other pirates die. how can pirate 6 save his skin?

thanks to my super smart professor brother

solution: pirates revisited

solution: pirates revisited

problem: a slightly different version of the original pirates problem (read that one first to get all the rules). 6 pirates, only one gold coin. as before, the pirates are super-smart, and they value, in this order: (i) their lives, (ii) getting money, (iii) seeing other pirates die. so if given the choice between two outcomes, in which they get the same amount of money, they'd choose the outcome where they get to see more of the other pirates die. how can pirate 6 save his skin?1 pirate case is obvious. 2 pirate case is obvious, pirate 2 keeps the coin.

3 pirate case: pirate 3 has to give the coin to pirate 1, because if he gives it pirate 2 pirate 2 will say "screw that i wanna see you die and i'm going to get the coin anyway." so in 3 pirate case, we have pirate 1 gets 1 coin, pirates 2 and 3 get 0.

4 pirate case: pirate 4 can't give the coin to pirate 1, because pirate 1 would rather see him die since he's going to get 1 coin anyway. But pirate 4 could give the coin to either pirate 2 or pirate 3.

5 pirate case: pirate 5 just dies, and it goes to the 4 pirate case. there is no way for him to convince two people to vote for him.

6 pirate case: pirate 6 can count on his own vote, plus pirate 5's vote, because 5 won't want to die. now who should he give the coin to? he could give it to pirate 1 or to pirate 4, since in the 4 pirate case they are guaranteed to get nothing. it's unclear whether he could give it to pirate 2 or 3. neither pirate 2 nor pirate 3 is guaranteed to get a coin in the 4 pirate case. so the question is, how do they value (i) definitely getting a coin from pirate 6 vs. (ii) definitely seeing pirates 6 and 5 die, with the chance of getting a coin from pirate 4. since we don't have enough information to answer that, to be safe, i would just say pirate 6 should offer the coin to pirate 1 or pirate 4.

fuse on fire aha:!!

a mad bomber is out on the job, making bombs. he has two fuses (pieces of string) of varying thickness which each burn for 30 seconds. unfortunately he wants this bomb to go off in 45 seconds. he can't cut the one fuse in half because the fuses are different thicknesses and he can't be sure how long it will burn. how can he arrange the fuses to make his bomb go off at the right time?

solution: fuse on fire

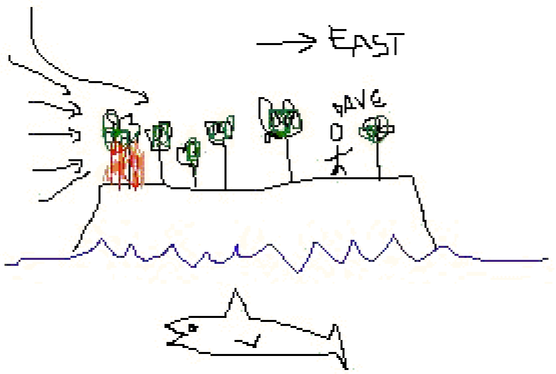
solution: fuse on fire

problem: a mad bomber is out on the job, making bombs. he has two fuses (pieces of string) of varying thickness which each burn for 30 seconds. unfortunately he wants this bomb to go off in 45 seconds. he can't cut the one fuse in half because the fuses are different thicknesses and he can't be sure how long it will burn. (for example: the first half of the fuse might burn up in 10 seconds and the second half in 20 seconds.. the fuse doesn't burn at a constant rate, but the total time it would burn is 30 seconds). how can he arrange the fuses to make his bomb go off at the right time?

solution: light both ends of one of the fuses. when that fuse goes out, 15 seconds has elapsed. then light the other fuse.

dave's on fire aha:

dave winer is stuck on a deserted island, with lots of trees, which is very thin and ten miles long (east to west). large cliffs surround the entire island and if he jumped off, he wouldn't survive the fall. a fire starts burning at the west side of the island. unfortunately this island always has a west to east blowing wind blowing at 2mph and this moves the fire slowly toward dave at 1mph. (so he only has ten hours left). save dave (or maybe, let him burn :-) ! what to do?



solution: dave's on fire

Solved by michael on February 27, 2001

solution: dave's on fire

someone suggested he could dig a pit across the island to act as a firebreak. good suggestion, if he had a shovel and the ground wasn't too hard.

but even if he didn't have a shovel, he could pick up a branch and run up to the fire and light the branch. then run all the way to the eastern edge of the island, but stop about a mile short. there he could light all those trees on fire and they would start burning and the fire would move east. it would consume all that vegetation in an hour, and then dave could wait for awhile for that part to cool down some. when the initial fire reached him, he could just run into the already burnt part, and the fire couldn't get him.

T = tree
D = dave
B = burnt

```
top view
==========================
| fire-> T T T T T T T D |
===========================
```

dave gets branch and lights it from fire

```
===========================
| fire-> D T T T T T T T |
===========================
```

dave lights trees farther east on the island

```
===========================
| fire-> T T T T D fire->|
===========================
```

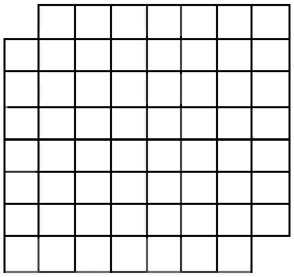
dave waits until second fire cools, and then hides out there

```
===========================
| B B B fire-> T T B B D |
===========================
```

dave is saved!

chessboard aha:!!!

problem: using 31 dominoes, where one domino covers exactly two squares, can you cover all the empty squares on this chessboard (which has 62 spaces). if so, how? if not, why?

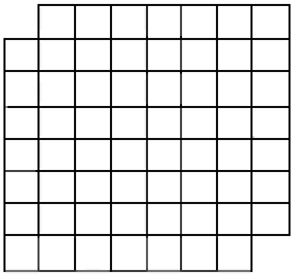


solution: chessboard

Solved by michael on March 1, 2001

solution: chessboard

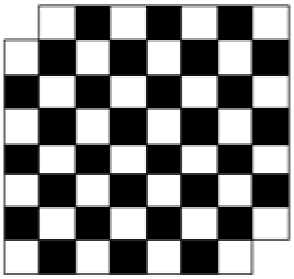
problem: using 31 dominoes, where one domino covers exactly two squares, can you cover all the empty squares on this chessboard (which has 62 spaces). if so, how? if not, why?



solution: i think everyone's first inclination is to try and figure out how it is possible. then again, if you've heard a bunch of these questions before, you usually know that if the question says "if not, why?" or "prove whether its possible or impossible", you can infer that it is not possible (otherwise, the question usually just asks for the solution).

after awhile fiddling around with the dominoes, you will start to realize that the most obvious solutions are not viable. if you start to think its impossible, think about why.

a reader emailed me telling me that using the picture of the chessboard without the squares colored will result in a very small percentage of people solving the problem. if i use the following picture



a much higher percentage of people will be able to solve the problem. the people who solve the problem using only the grid, usually represent the problem in their heads as the colored board.

still can't figure it out? using the colored board, remember that a domino always covers a black square and a white square.

if you look at the board, you will see that the two squares missing are both black. this means that for all the squares on the board, each white square has a corresponding black square, except for two white ones. so even if you covered all the black/white pairs, you would still be left with two white squares, which will never be adjacent to each other, no matter how you lay out the dominoes.

easy river crossing aha:!

three cannibals and three anthropologists have to cross a river. the boat they have is only big enough for two people. if at any point in time there are more cannibals on one side of the river than anthropologists, the cannibals will eat them. what plan can the anthropologists use for crossing the river so they don't get eaten?

remember! the boat can't cross the river by itself, someone has to be in it to row it across.

a much harder river crossing problem will appear later this week.

```
A - anthropologist
C - cannibal
++ - boat

      river
AAA |============|
   |++         |
CCC |============|

need to make it
      river
   |============| AAA
   |         ++|
   |============| CCC
```

note that if you violate the "anthropologists > cannibals" rule at any point in time, it is illegal.. for example if a boat with a cannibal and an anthropologist travels to a shore with one cannibal on it, then # cannibals > # anthropologists, even if you say the anthropologist immediately takes the boat back.

solution: easy river crossing

solution: easy river crossing

problem: three cannibals and three anthropologists have to cross a river. the boat they have is only big enough for two people. if at any point in time there are more cannibals on one side of the river than anthropologists, the cannibals will eat them. what plan can the anthropologists use for crossing the river so they don't get eaten?

```
A - anthropologist
C - cannibal
++ - boat

      river
AAA |============|
   |++         |
CCC |============|
```

need to make it

```
        river
    |============| AAA
    |          ++|
    |============| CCC
```

note that if you violate the "anthropologists > cannibals" rule at any point in time, it is illegal.. for example if a boat with a cannibal and an anthropologist travels to a shore with one cannibal on it, then # cannibals > # anthropologists, even if you say the anthropologist immediately takes the boat back.

Let W be the west shore which they are all on. Let E be the east shore where they want to go.

1. A and C cross
W = { A, A, C, C }
E = { A, C }

2. A returns
W = { A, A, A, C, C }
E = { C }

3. C and C cross
W = { A, A, A }
E = { C, C, C }

4. C returns
W = { A, A, A, C }
E = { C, C }

5. A and A cross
W = { A, C }
E = { A, A, C, C }

6. A and C return
W = { A, A, C, C }
E = { A, C }

7. A and A cross
W = { C, C }
E = { A, A, A, C }

8. C returns
W = { C, C, C }
E = { A, A, A }

9. C and C cross
W = { C }
E = { A, A, A, C, C }

10. C returns
W = { C, C }
E = { A, A, A, C }

11. C and C cross
W = true
E = { A, A, A, C, C, C }

March 6, 2001

shapes aha:!!

part I: draw a square. divide it into four identical squares. remove the bottom left hand square. now divide the resulting shape into four identical shapes.

part II: draw an equilateral triangle (all sides same length). divide it into four identical shapes. remove the bottom left hand shape. now divide the resulting shape into four identical shapes.

this is the sort of problem that i would expect on a MENSA test. i'm not too sure whether getting this right constitutes intelligence in a way that would benefit

computer scientists, but maybe it does. if you figure it out, then you can say it does. if you can't figure it out, then you can just say it's all hogwash and it's a stupid question.

thanks to mark chesser

solution: shapes

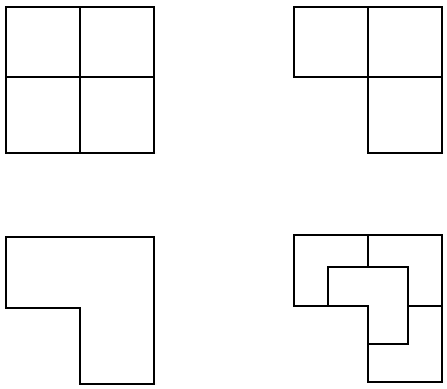
solution: shapes

part I: draw a square. divide it into four identical squares. remove the bottom left hand square. now divide the resulting shape into four identical shapes.

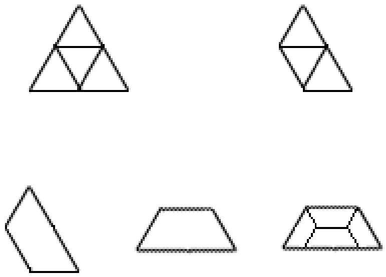
part II: draw an equilateral triangle (all sides same length). divide it into four identical shapes. remove the bottom left hand shape. now divide the resulting shape into four identical shapes.

this is the sort of problem that i would expect on a MENSA test. i'm not too sure whether getting this right constitutes intelligence in a way that would benefit computer scientists, but maybe it does. if you figure it out, then you can say it does. if you can't figure it out, then you can just say it's all hogwash and it's a stupid question.

solution:



part II



webloggers aha:!

five webloggers - joshua Allen, meg Hourihan, jason Kottke, robert Scoble, and joel Spolsky - were competing for karma points on the major search engines: google, yahoo, altavista, lycos, and msn. karma was distributed on a five point scale. the most popular weblog received 5 points, and the least popular received 1 point. for each search engine, no two webloggers received the same number of points. overall scores were determined by adding up the individual scores from each search engine.

Allen got the highest number of karma points - 24. Kottke was consistent in his scores: he got the same karma points from 4 different search engines. Spolsky got 5 points from lycos, and 3 from msn.

no two webloggers got the same total score, and the final rankings were as follows: Allen, Hourihan, Kottke, Scoble, and Spolsky. how many karma points did Hourihan get from lycos?

solution: webloggers

solution: webloggers

five webloggers - joshua Allen, meg Hourihan, jason Kottke, robert Scoble, and joel Spolsky - were competing for karma points on the major search engines: google, yahoo, altavista, < a>lycos, and msn. karma was distributed on a five point scale. the most popular weblog received 5 points, and the least popular received 1 point. for each search engine, no two webloggers received the same number of points. overall scores were determined by adding up the individual scores from each search engine.

Allen got the highest number of karma points - 24. Kottke was consistent in his scores: he got the same karma points from 4 different search engines. Spolsky got 5 points from lycos, and 3 from msn.

no two webloggers got the same total score, and the final rankings were as follows: Allen, Hourihan, Kottke, Scoble, and Spolsky. how many karma points did Hourihan get from lycos?

solution: let's start with what we know

	G	Y	A	L	M	Total
A						24
H						
K						
Sc						
Sp				5	3	

the only possible values for Allen achieving 24 is { 5 5 5 5 4 } and since Spolsky got a 5 from lycos, we know that is where Allen's 4 comes from.
we also know that the total number of points given out was 75.
(5 * (5 + 4 + 3 + 2 + 1))
spolsky had to have at least 11 points. if Spolsky had more than 11 points, say 12, then is it possible to achieve a solution? Scoble would have had to have at least 13 (since there were no ties), and Kottke 14, and Houlihan 15. that would yield an overall total of 78. too much! so Spolsky definitely had 11 points.

	G	Y	A	L	M	Total
A	5	5	5	4	5	24
H						
K						
Sc						
Sp	1	1	1	5	3	11

using the same logic as before, we also know that Scoble could not have gotten more than 12 points. if he had 13, and Kottke 14, and Houlihan 15, the total would be 77. still too much. so Scoble had 12, and continuing on Kottke had to have 13 and Houlihan 15, otherwise the totals would be over 75.

now we know Kottke had 14 points. if he got four 4's for consistency, it wouldn't work (already over 16). if he got four 2's, it also wouldn't work (8 points plus the maximum 5 is still only 13). so he had to have received four 3's. and since he couldn't have gotten a 3 from msn, that is where he received a 1.

	G	Y	A	L	M	Total
A	5	5	5	4	5	24
H						15
K	3	3	3	3	1	13
Sc						12
Sp	1	1	1	5	3	11

let's look at scoble. we can see from the chart that all 5's and 3's have already been given out (and there is only one 1 left). so Scoble's scores can only contain 4's, 2's, or a single 1. given that information the only possible combination of 5 scores that would yield 12 is { 2 2 2 2 4 }. since Allen already has a 4 from lycos, Scoble must have a 2 there.

	G	Y	A	L	M	Total
A	5	5	5	4	5	24
H						15
K	3	3	3	3	1	13
Sc				2		12
Sp	1	1	1	5	3	11

hence Houlihan must have a 1 from lycos!

	G	Y	A	L	M	Total
A	5	5	5	4	5	24
H				**1**		15
K	3	3	3	3	1	13
Sc				2		12
Sp	1	1	1	5	3	11

March 9, 2001

hard river crossing aha:!

a disfunctional family has to cross the river. on one side of the river are a mom and 2 daughters, dad and 2 sons, the maid and the dog. there is a boat only big enough to hold 2 people (counting the dog as 1 person). only the adults are capable of operating the boat. everyone has to get to the other side, without anything bad happening.

difficulties: if the dog is left with anyone and the maid isn't there to control him, he'll bite. the dad can't be left with any of the daughters when the mom isn't there. likewise, the mom can't be trusted alone with either of the sons when the dad isn't there.

remember! only an adult can operate the boat, AND the boat can't drive itself.

solution: hard river crossing

solution: hard river crossing

a disfunctional family has to cross the river. on one side of the river are a mom and 2 daughters, dad and 2 sons, the maid and the dog. there is a boat only big enough to hold 2 people (counting the dog as 1 person). only the adults are capable of operating the boat. everyone has to get to the other side, without anything bad happening.

difficulties: if the dog is left with anyone and the maid isn't there to control him, he'll bite. the dad can't be left with any of the daughters when the mom isn't there. likewise, the mom can't be trusted alone with either of the sons when the dad isn't there.

remember! only an adult can operate the boat, AND the boat can't drive itself.

solution:
we start with a mother (m), two daughters (d1, d2), a father (f), two sons (s1, s2), a housemaid (h), and a dog (c - canine) on the west (W) shore, and they all want to get to the east (E) shore.

W = {m, d1, d2, f, s1, s2, h, c} // everyone on the west shore
E = {} // no one on the east shore
let's move everyone, over...
housemaid and canine go east, and the housemaid comes back:
W = {m, d1, d2, f, s1, s2, h}
E = {c}
housemaid and s1 go east, h and c come back:
W = {m, d1, d2, f, s2, h, c}
E = {s1}
father and s2 go east, father comes back:
W = {m, d1, d2, f, h, c}
E = {s1, s2}
mother and father go east, mother comes back:
W = {m, d1, d2, h, c}
E = {f, s1, s2}
h and c go east, father comes back:
W = {m, d1, d2, f}
E = {s1, s2, h, c}
father and mother go east, mother comes back:
W = {m, d1, d2}
E = {f, s1, s2, h, c}
mother and d1 go east, housemaid and c come back:
W = {d2, h, c}
E = {m, d1, f, s1, s2}
h and d2 go east, h comes back
W = {h, c}
E = {m, d1, d2, f, s1, s2}

h and c go east
W = {}
E = {m, d1, d2, f, s1, s2, h, c}
done!
thanks babak for the writeup.

classic weighing aha:!

this is a classic problem which i have heard many times before. this is the "harder" of the two problems, since in this one, you do not know if the invalid item weighs more or less than the others.

solving it is only half the battle. writing up a solution that anyone including your grandma could understand, is very hard.

problem: the evil king from before sends his own assassin to take care of the evil queen who tried to poison him. of course, her trusty guards catch the assassin before any harm is done. the queen notices that the assassin is quite handsome and doesn't really want to punish him by death. she decides to test his wisdom.

the queen gives the assassin 12 pills which are all completely identical in shape, smell, texture, size, except 1 pill has a different weight. the queen gives the man a balance and tells him that all the pills are deadly poison except for the pill of a different weight. the assassin can make three weighings and then must swallow the pill of his choice. if he lives, he will be sent back to the bad king's kingdom. if he dies, well, thats what you get for being an assassin.

only one pill is not poison and it is the pill which has a different weight. the assassin does not know if it weighs more or less than the other pills. how can he save his skin?

solution: classic weighing

solution: classic weighing

this is a classic problem which i have heard many times before. this is the "harder" of the two problems, since in this one, you do not know if the invalid item weighs more or less than the others.

solving it is only half the battle. writing up a solution that anyone including your grandma could understand, is very hard.

problem: the evil king from before sends his own assassin to take care of the evil queen who tried to poison him. of course, her trusty guards catch the assassin

before any harm is done. the queen notices that the assassin is quite handsome and doesn't really want to punish him by death. she decides to test his wisdom.

the queen gives the assassin 12 pills which are all completely identical in shape, smell, texture, size, except 1 pill has a different weight. the queen gives the man a balance and tells him that all the pills are deadly poison except for the pill of a different weight. the assassin can make three weighings and then must swallow the pill of his choice. if he lives, he will be sent back to the bad king's kingdom. if he dies, well, thats what you get for being an assassin.

only one pill is not poison and it is the pill which has a different weight. the assassin does not know if it weighs more or less than the other pills. how can he save his skin?

solution: easy.

choose any eight of the pills and put four of them on each side of the balance.

there are two possibilities:

(1) one side of the balance comes out lighter. In this case, you know that the abnormal (safe) pill is one of the pills already on the balance. label the pills on the lighter side A B C and D, and the pills on the heavier side E F G and H. label the pills not on the balance NORM (you know they are normal pills).

(2) the balance is even. in this case, you know that the abnormal (safe) pill is one of the pills not on the balance. label the pills already on the balance NORM, and label the four pills not on the balance I J K and L.

lets proceed with possibility (1).

consider why the side ABCD came out higher than the side EFGH. this could be because:

A is the abnormal pill, and its lighter than the other pills.

B is the abnormal pill, and its lighter than the other pills.

C is the abnormal pill, and its lighter than the other pills.

D is the abnormal pill, and its lighter than the other pills.

E is the abnormal pill, and its heavier than the other pills.

F is the abnormal pill, and its heavier than the other pills.

G is the abnormal pill, and its heavier than the other pills.

H is the abnormal pill, and its heavier than the other pills.

now lets make another weighing, with two of the ABCD pills on either side, and one of the EFGH pills on either side. for example, lets weigh ABE versus CDF. how would this weighing come out given each of those 8 possibilities we just listed?

if A is the light pill, the ABE/CDF weighing will come out with ABE high.

if B is the light pill, the ABE/CDF weighing will come out with ABE high.

if C is the light pill, the ABE/CDF weighing will come out with ABE low.

if D is the light pill, the ABE/CDF weighing will come out with ABE low.

if E is the heavy pill, the ABE/CDF weighing will come out with ABE low.

if F is the heavy pill, the ABE/CDF weighing will come out with ABE high.

if G is the heavy pill, the ABE/CDF weighing will come out even.

if H is the heavy pill, the ABE/CDF weighing will come out even.

OK, so we observe how the ABE versus CDF weighing actually comes out.

(a) if it comes out even, then we know that the abnormal pill is either G or H. for our third weighing, we can weigh G against one of the pills we already know to be normal (one of the pills we labelled NORM). if it comes out even, then G is normal and H must be the abnormal pill. if it comes out uneven, then G is the abnormal pill.

(b) as we can see from our chart above, if the ABE/CDF weighing comes out with ABE high, then the situation is either: A is the light pill, B is the light pill, or F is the heavy pill.

(c) as we can see from our chart above, if the ABE/CDF weighing comes out with ABE low, then the situation is either: C is the light pill, D is the light pill, or E is heavy pill.

so in either situation (b) or (c), we have two possible light pills and one possible heavy pill. what we do in that case is we put one of the possible light pills and the possible heavy pill on one side of the scale, and two NORM pills on the other side of the scale. this is our third weighing. if it comes out even, then we know that the other possible light pill is the abnormal pill. if it comes out with the two

NORM pills high, then we know that one of the pills on the other side is abnormally heavy, so we know that the possible heavy pill is the culprit. if it comes out with the two NORM pills low, then we know that one of the pills on the other side is abnormally light, so we know that the possible light pill on the scale is the culprit.

that takes care of case (1), where the first weighing came out uneven.

what about case (2), where the first weighing comes out even?

then we know the abnormal pill is one of I J K or L, and we have two weighings to find the abnormal pill in.

for our second weighing, we put I and J on one side of the scale, and two NORM pills on the other.

(a) if this comes out uneven, we know the abnormal pill is I or J; we weigh I against one NORM pill to see if I is abnormal and if it isnâ€™t, we can conclude that J is the abnormal pill.

(b) if the IJ versus 2 NORM weighing comes out even, we know the abnormal pill is K or L; we weight K against one NORM pill to see if K is abnormal and if it isnâ€™t, we can conclude that L is the abnormal pill.

finished.

monty hall problem aha:!

another well known problem in probability is the monty hall problem.

you are presented with three doors (door 1, door 2, door 3). one door has a million dollars behind it. the other two have goats behind them. you do not know ahead of time what is behind any of the doors.

monty asks you to choose a door. you pick one of the doors and announce it. monty then counters by showing you one of the doors with a goat behind it and asks you if you would like to keep the door you chose, or switch to the other unknown door.

should you switch? if so, why? what is the probability if you don't switch? what is the probability if you do.

lots of people have heard this problem.. so just knowing what to do isn't sufficient. its the explanation that counts!

solution: monty hall

solution: monty hall problem

another well known problem in probability is the monty hall problem.

you are presented with three doors (door 1, door 2, door 3). one door has a million dollars behind it. the other two have goats behind them. you do not know ahead of time what is behind any of the doors.

monty asks you to choose a door. you pick one of the doors and announce it. monty then counters by showing you one of the doors with a goat behind it and asks you if you would like to keep the door you chose, or switch to the other unknown door.

should you switch? if so, why? what is the probability if you don't switch? what is the probability if you do.

lots of people have heard this problem.. so just knowing what to do isn't sufficient. its the explanation that counts!

the answer is that yes, you should *always* switch as switching increases your chances from 1/3 to 2/3. how so, you ask? well, lets just enumerate the possibilities.

	door 1	door 2	door 3
case 1	$$	goat	goat
case 2	goat	$$	goat
case 3	goat	goat	$$

its clear that if you just choose a door and stick with that door your chances are 1/3.
using the switching strategy, let's say you pick door 1. if its case 1, then you lose. if it's case 2, monty shows you door 3, and you switch to door 2, you win. if it's case 3, monty shows you door 2, and you switch to door 3, you win. it doesn't matter what door you pick in the beginning, there are always still three possibilities. one will cause you to lose, and two will cause you to win. so your chances of winning are 2/3.
the solution all resides in the fact that monty knows what is behind all the doors and therefore always eliminates a door for you, thereby increasing your odds.
maybe its easier to see in this problem. there are 1000 doors, only one of which has a prize behind it. you pick a door, then monty opens 998 doors with goats behind them. do you switch? it seems more obvious in this case, because monty had to take care in which door not to open, and in the process basically showing you where the prize was (999 out of 1000 times).

gold chain aha:!!!

a man has a gold chain with 7 links. he needs the service of a laborer for 7 days at a fee of one gold link per day. however, each day of work needs to be paid for separately. in other words, the worker must be paid each day after working and if the laborer is ever overpaid he will quit with the extra money. also he will never allow himself to be owed a link.

what is the fewest # of cuts to the chain to facilitate this arrangement and how does that guarantee payment?

solution: gold chain

solution: gold chain

Can we get change back from the laborer?

If so, we cut one link to make a chain of 4 links, a chain of 2 links and the cut link itself.

Day 1, we give him the cut link
Day 2, we take back the cut link, give him the 2 link chain
Day 3, we give him the cut link
Day 4, we take back both the cut link and the 2 link chain, give him the 4 link chain
Day 5, we give him the cut link
Day 6, we take back the cut link, give him the 2 link chain
Day 7, we give him the cut link

surgeons aha:!!

a one armed surgeon with a hand wound needs to operate on three patients. the surgeon only has two gloves. how can he operate on the three patients in turn without risking exchange of fluids? (remember he only has one arm so he only needs to wear one glove at a time.)

solution: surgeons

solution: surgeons

problem: a one armed surgeon with a hand wound needs to operate on three patients. the surgeon only has two gloves. how can he operate on the three patients in turn without risking exchange of fluids? (remember he only has one arm so he only needs to wear one glove at a time.)

solution: the surgeon places both gloves on his hand (1 and 2). he operates on patient A. he then takes the top glove off (#2), leaving on the bottom glove (#1) and operates on patient B. then he carefully reverses glove #2, so the clean side is on the outside, and he places it on top of glove #1 which is on his hand, and operates on patient C.

this problem is kind of dumb because how's the surgeon going to change the gloves on his hand when he only has one hand. plus no offense, but how often do you come across a one-armed surgeon (i'm sure there are plenty of one-armed doctors, but a surgeon!?!). anyway, i had to make this problem child friendly and changing the story to the above was the only way to do it. consider for a minute what the initial problem was. the surgeon was just a guy, the patients were women, and the glove was... well, i won't insult your intelligence.

clock aha:!

part I: what is the angle between the minute hand and the hour hand at 3:15 on an analog clock? no, its not 0.

part II: how often does the minute hand pass the hour hand on an analog clock?

solution: clock

solution: clock

part I: what is the angle between the minute hand and the hour hand at 3:15 on an analog clock? no, its not 0.

part II: how often does the minute hand pass the hour hand on an analog clock?

answer: part I: 12 hours on the clock make 360 deg. so one hour is 30 deg. the hour hand will be directly on the 3 when the minute hand is at 12 (3:00). after 15 minutes or 1/4 of an hour, the hour hand will be 1/4 * 30 deg = 7.5 deg. away from the minute hand.

part II: if you just think about it, intuitively you'll see the minute hand passes the hour hand 11 times every 12 hours, so it must pass it every 1 1/11 hours. but this doesn't make sense to me. i need to prove it.

if x is our answer then every x hours, the minute hand and the hour hand will be right on top of each other. every hour the hour hand travels 5 units. so between every time that the minute and the hour hand meet, the hour hand will go 5*x units. every hour the minute hand travels 60 units, so it will have gone 60*x units.

what we're trying to find is the distance traveled by the minute hand to reach the hour hand, once the minute hand has looped around once. consider its 12:00. both hands in the same position. after an hour, minute hand is on 12, hour hand on 1 (its traveled 5 units). now in the time it takes the minute hand to catch up to the hour hand it will travel a little bit further.

we only need to find x where 5*x = 60*(x-1), since the real distance traveled by the minute hand, from where it started to where it ends, is 60*(x-1). the first hour just puts it back where it started, so we're only concerned with the extra part it traveled to reach the hour hand.

$5x = 60(x-1)$
$5x = 60x - 60$
$60 = 55x$
$60/55 = x$
there it is. the answer is 60/55 hours, or every 1 and 1/11 hours.
i apologize that this is horribly confusing, but if you stare at it long enough it will make sense.

mountain man aha:!

at 6 a.m. a man starts hiking a path up a mountain. he walks at a variable pace, resting occasionally, but never actually reversing his direction. at 6 p.m. he reaches the top. he camps out overnight. the next morning he wakes up at 6 a.m. and starts his descent down the mountain. again he walks down the path at a variable pace, resting occassionally, but always going downhill. at 6 p.m. he reaches the bottom. what is the probability that at some time during the second day, he is in the exact same spot he was in on the first day?

solution: mountain man

solution: mountain man

problem: at 6 a.m. a man starts hiking a path up a mountain. he walks at a variable pace, resting occasionally, but never actually reversing his direction. at 6 p.m. he reaches the top. he camps out overnight. the next morning he wakes up at 6 a.m. and starts his descent down the mountain. again he walks down the path at a variable pace, resting occassionally, but always going downhill. at 6 p.m. he reaches the bottom. what is the probability that at some time during the second day, he is in the exact same spot he was in on the first day?

answer: the probability is 100%. the easiest way to see it is, consider that on the second day when the man is going down the mountain, a ghost follows his original pace up the mountain. so even if he varies his pace as he goes down the mountain, at some point in time, he will be in the same spot as the ghost, and therefore, the same spot he was in the day before.

treasure island aha:!

you find an old treasure map in your grandma's attic. the map shows a cannon, a coconut tree, and a palm tree. the map states that to find the treasure you must:
a. start at the cannon, walk toward the palm tree while counting your paces. when you reach the palm tree, turn 90 degrees to your **left** and walk the same number of paces. mark that spot on the ground with a stake.
b. start at the cannon again, walk toward the coconut tree while counting your steps. when you reach the coconut tree, turn 90 degrees to your **right** and walk the same number of paces. mark that spot on the ground with a stake.
c. find the midpoint between the two stakes and dig for the treasure.

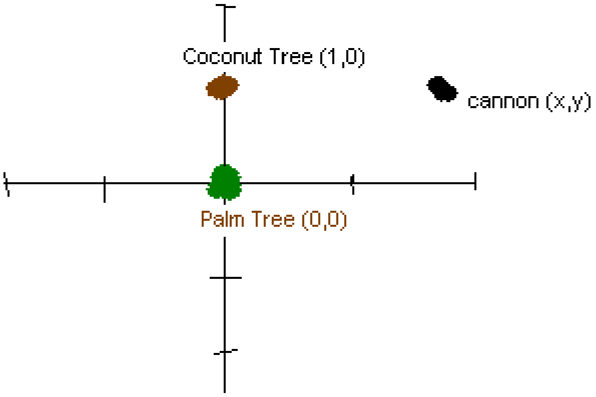
you set off in secrecy to the deserted island. upon reaching the shore you site the coconut tree and the palm tree, but someone has removed the cannon. without digging randomly all over the island, is it still possible to find the treasure?

solution: treasure island

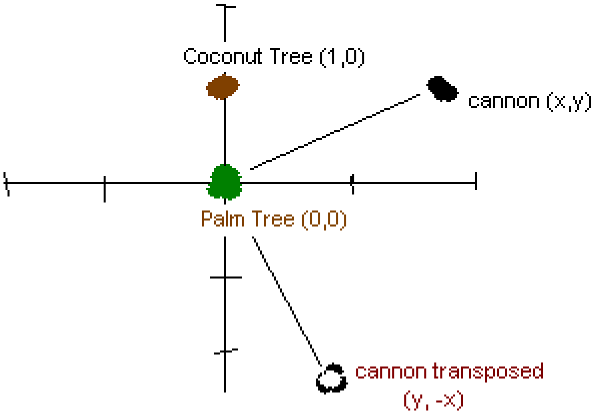
problem: you find an old treasure map in your grandma's attic. the map shows a cannon, a coconut tree, and a palm tree. the map states that to find the treasure you must:
a. start at the cannon, walk toward the palm tree while counting your paces. when you reach the palm tree, turn 90 degrees to your **left** and walk the same number of paces. mark that spot on the ground with a stake.
b. start at the cannon again, walk toward the coconut tree while counting your steps. when you reach the coconut tree, turn 90 degrees to your **right** and walk the same number of paces. mark that spot on the ground with a stake.
c. find the midpoint between the two stakes and dig for the treasure.

you set off in secrecy to the deserted island. upon reaching the shore you site the coconut tree and the palm tree, but someone has removed the cannon. without digging randomly all over the island, is it still possible to find the treasure?

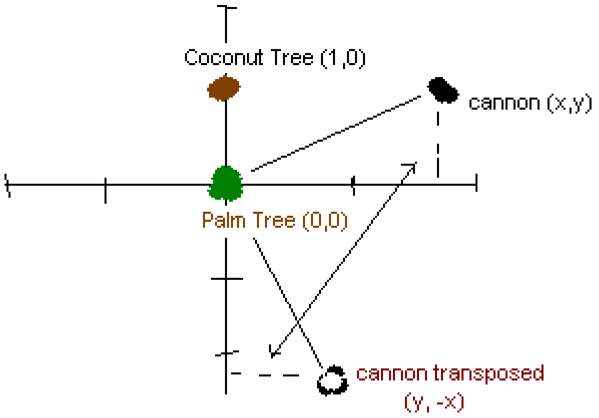
solution: this just takes basic geometry skills. when we get to the island all we see are the coconut and palm trees. so lets lay out our coordinate system such that the palm tree is at (0,0) and the coconut tree is at (1,0). it honestly doesn't matter how you describe the coordinate system - you could say the coconut is at (c,0) if you like, or even (0,c) or (0,1). we are just placing our own coordinate system on top of the existing surface. if you use a different system, you will get a different numerical answer but the same positional answer in the real world.



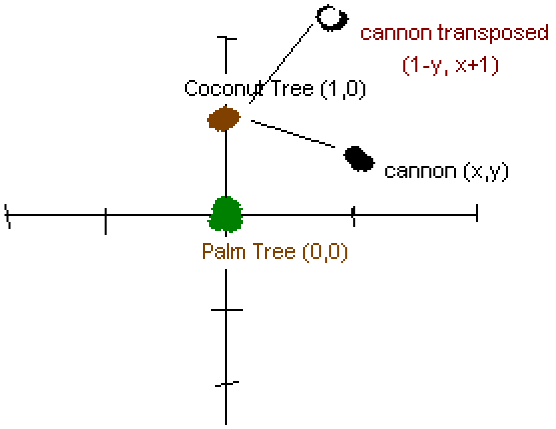
here is our island. the cannon is at (x,y) because we have no idea where it is, so x and y are the unknowns. (note the cannon doesn't have to be in the upper right quadrant but it won't make a difference in the solution because x and y could be negative if we want them to be).



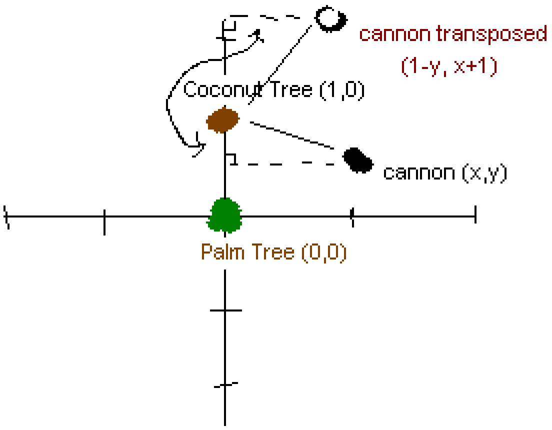
if we walk to the palm and turn right we can easily see the way it lays out in the diagram. you basically have just transposed the x and y positions.



it might be easier to think of it as a triangle, and flipping the triangle around to find the point for the next stake at (y, -x)



we do the same thing with the coconut tree, although here its only a tiny bit trickier because we have to factor in the position of the coconut tree at (1,0) or (c,0).



we use the idea of a triangle to help us better understand that the stake will end up at (1-y, x+1) or (c-y, x+c).

then to find the midpoint of the two points which is just the first position + the second position divided by two.

(y, -x) and (1-y, x+1)
((y - y + 1)/2, (x - x + 1)/2)
(1/2, 1/2)
hence our answer is (1/2,1/2) - although we'll see that if we had use the constant C for the coconut tree we would have ended up with (c/2, c/2). this is important because even though we laid out the island in our own coordinate system, its not always the case that c must be positive. i think (and i've seen this solution elsewhere, but can't really come up with a repro, so let me know if i'm wrong here) there really are two places the treasure could be because c could also be negative. so if we use our first answer of (1/2,1/2) we must also search at (-1/2, -1/2). if the cannon was in the lower half of the coordinate system you can see that the treasure would actually be in that quadrant also.

screwy pirates aha:!

a. first i would just like to say i am so glad jenna got voted off survivor. yuck, i couldn't stand her ever since she accused that guy of eating beef jerky when he was chewing on grass.
b. recruit meyer on boot camp rocks. did you see how he made himself cry to get sympathy? and it worked! he is the funniest part of the show. i hope he wins.
c. i apologize for this site being **soooooo slow**. but that's why you get what you pay for. (i didn't pay anything so i shouldn't really expect anything in return.)

d. those screwy pirates are at it again. this time there are 13 pirates and they need to protect their treasure chest. they decide that they should only be able to open the chest if the majority (at least 7) agree that it should be opened. they ask a locksmith to come and put a specific number of locks on the safe. every lock must be opened to open the chest. there can be multiple keys for each lock, but each key only opens one lock (i.e. no skeleton keys). the locksmith can give more than one key to each pirate. how many locks should the locksmith use and what strategy should he use to distribute the keys, such that only when a majority of the pirates agree can the chest be opened?

solution: screwy pirates

solution: screwy pirates aha:

problem: those screwy pirates are at it again. this time there are 13 pirates and they need to protect their treasure chest. they decide that they should only be able to open the chest if the majority (at least 7) agree that it should be opened. they ask a locksmith to come and put a specific number of locks on the safe. every lock must be opened to open the chest. there can be multiple keys for each lock, but each key only opens one lock (i.e. no skeleton keys). the locksmith can give more than one key to each pirate. how many locks should the locksmith use and what strategy should he use to distribute the keys, such that only when a majority of the pirates agree can the chest be opened?

a somewhat technical answer is here.

(un)fortunately that user left it up to me to explain why this solution works... but fortunately my brother did it here in a very understandable fashion. thanks big brother!

technical solution for screwy pirates

Here is solution that works for any number of pirates and any number of pirates needed to open to chest.

Let T be the total number of pirates. Let N be the number of pirates required to open the chest.

The number of locks needed would be L = (T,N-1) = (T!)/[(N-1)! * (T-N-1)!]. The number of keys each pirate would have be K = (T-1,N-1) = (T-1)!/[(N-1)! * (T-N)!].

For this specific problem, the number of locks would be 1716 and the number of keys per pirate would be 924.

I know I haven't provided an explanation of why and how this system works. We'll leave that lengthy, involved explanation to the author.

Note: the notation (x,y) is the combination notation; I can't use the conventional combination notation in plain text. (x,y) basically asks the question "how many ways can you pick y objects from a group of x objects?" (x,y) = (x!)/[y! * (x-y)!]

easy solution for screwy pirates

Here's a less elegant solution. How many different combinations of 6 pirates are there? 13 choose 6 is: 13!/6!7! That's a hell of a lot of locks. But you could say, put that many locks on the chest. And for each lock, you distribute keys to all the pirates EXCEPT the selection of 6 corresponding to that lock. Then any selection of 7 pirates is guaranteed to be open all the locks. For consider the first 6 of them. There will be exactly ONE lock that those six are unable, between the 6 of them, to open. But the seventh pirate will be able to open that lock. Because keys to that lock went out to everybody except those 6.

Also, any selection of 6 pirates is guaranteed to encounter exactly one lock that they are unable to open. (So any selection of fewer than 6 is guaranteed to encounter 1 or more locks they are unable to open...)

So that will do the trick, and the explanation of why it would do the trick is elegant. But I don't feel that it's an elegant solution, because of there being so many locks... Anyone have something better?

ants on a triangle aha:!

there are three ants on a triangle, one at each corner. at a given moment in time, they all set off for a different corner at random. what is the probability that they don't collide?

solution: ants on a triangle

solution: ants on a triangle

Consider the triangle ABC. We assume that the ants move towards different corners along the edges of the triangle.

Total no. of movements: 8 A->B, B->C, C->A A->B, B->A, C->A A->B, B->A, C->B A->B, B->C, C->B A->C, B->C, C->A A->C, B->A, C->A A->C, B->A, C->B A->C, B->C, C->B

Non-colliding movements: 2 A->B, B->C, C->A A->C, B->A, C->B

(i.e. the all ants move either in the clockwise or anti-clockwise direction at the same time)

P(not colliding) = 2/8 = 0.25

one mile south aha:!!

how many places are there on the earth that one could walk one mile south, then one mile east, then one mile north and end up in the same spot? to be precise, let's assume the earth is a solid smooth sphere, so oceans and mountains and other such things do not exist. you can start at any point on the sphere and walk in any direction you like.

think you've figured it out? i'll tell you now, there is more than one. in fact, there are more than two. also be advised that walking north from the north pole (or south from the south pole) is illogical and therefore does not enter into the problem. all normal assumptions about directions will be used.

there are **no** tricks involved with this question. it just forces you to really think about the problem to come up with all the solutions.

solution: one mile south

November 14, 2001

solution: one mile south aha:

problem: how many places are there on the earth that one could walk one mile south, then one mile east, then one mile north and end up in the same spot? to be precise, let's assume the earth is a solid smooth sphere, so oceans and mountains and other such things do not exist. you can start at any point on the sphere and walk in any direction you like. think you've figured it out? i'll tell you now, there is more than one. in fact, there are more than two. also be advised that walking north from the north pole (or south from the south pole) is illogical and therefore does not enter into the problem. all normal assumptions about directions will be used.

there are no tricks involved with this question. it just forces you to really think about the problem to come up with all the solutions.

solution:

well the north pole is one such place.

then somewhere near the south pole such that when you walk one mile south you are at the point on the earth where the circumference is 1. that way when you walk 1 mile east, you end up back at the same point. and of course one mile north from there puts you back where you started. here is a drawing courtesy of jy. there may or may not be such a place in the northern hemisphere where walking a mile south puts you at the 1 mile circumference point on the earth.

i'm no geometry sphere expert, so someone will have to let me know if that is physically possible (i.e. i tend to think that if you walk n units south from any point on the northern part of a sphere, other than the north pole, it is impossible for the circumference to be n or less than n, but who knows?)

finally there are actually an infinite number of points. if we consider the case before where we went to the point with a circumference of 1, why not go to the point with a circumference of 1/2. then when you go a mile east, you loop around twice, and end up in the same spot. this holds true for 1/3, 1/4, 1/5, ... 1/n, etc.

cube aha:!

this is difficult to describe in words, so read this carefully, lest there be any confusion. you have a normal six sided cube. i give you six different colors that you can paint each side of the cube with (one color to each side). how many different cubes can you make?

different means that the cubes can not be rotated so that they look the same. this is important! if you give me two cubes and i can rotate them so that they appear identical in color, they are the same cube.

solution: cube

solution: cube

Let X be the number of "different" cubes (using the same definition as in the problem). Let Y be the number of ways you can "align" a given cube in space such that one face is pointed north, one is east, one is south, one is west, one is up, and one is down. (We're on the equator.) Then the total number of possibilities is X * Y. Each of these possibilities "looks" different, because if you could take a cube painted one way, and align it a certain way to make it look the same as a differently painted cube aligned a certain way, then those would not really be different cubes. Also note that if you start with an aligned cube and paint it however you want, you will always arrive at one of those X * Y possibilities.

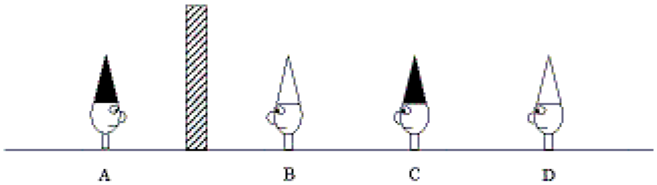
How many ways can you paint a cube that is already "aligned" (as defined above)? You have six options for the north side, five options for the east side, etc. So the total number is 6! (that's six factorial, or 6 * 5 * 4 * 3 * 2 * 1). Note that each way you do it makes the cube "look" different (in the same way the word is used above). So 6! = X * Y.

How many ways can you align a given cube? Choose one face, and point it north; you have six options here. Now choose one to point east. There are only four sides that can point east, because the side opposite the one you chose to point north is already pointing south. There are no further options for alignment, so the total number of ways you can align the cube is 6 * 4.

Remember, Y is defined as the number of ways you can align the cube, so Y = 6 * 4. This gives us 6! = X * 6 * 4, so X = 5 * 3 * 2 * 1 = 30.

more hat puzzles aha:!

i buried four fishermen up to their necks in the sand on the beach at low tide for keeping their fishing spot a secret from me. i put a hat on each of their heads and told them that one of them must shout out the correct color of their own hat or they will all be drowned by the incoming tide. i give them 10 minutes to do this. fisherman A and B can only see the sand dune i erected. fisherman C can see that fisherman B has a white hat on. fisherman D can see that C has a black hat, and B has a white hat. the fisherman have been told that there are four hats, two white and two black, so they know that they must have either a white or a black hat on. who shouts out the color of their hat and how do they know?



solution: fishermen

solution: more hat puzzles

Fisherman C shouts out.

Fishermen A and B are in the same situation - they have no information to help them determine their hat colour so they can't answer. C and D realise this.

Fisherman D can see both B and C's hats. If B and C had the same colour hat then this would let D know that he must have the other colour.

When the time is nearly up, or maybe before, C realises that D isn't going to answer because he can't. C realises that his hat must be different to B's otherwise D would have answered. C therefore concludes that he has a black hat because he can see B's white one.

coin on a table aha:!!!

you die and the devil says he'll let you go to heaven if you beat him in a game. the devil sits you down at a round table. he gives himself and you a huge pile of quarters. he says "ok, we'll take turns putting quarters down, no overlapping allowed, and the quarters must rest on the table surface. the first guy who can't put a quarter down loses." the devil says he wants to go first.

being the smart programmer you are, you realize that if the devil goes first, he may automatically win. so you convince him to let you go first, which makes your day because you know you can't lose. what is your winning strategy?

thanks to ole eichhorn

solution: coin on a table

solution: coin on a table

Here's what I came up with:

First, put the first quarter exactly in the center of the (perfectly circular) table.

Next, for each quarter the opponent places, place one directly opposite it. That is, place it so that the center of the table is halfway between your piece and the opponent's previous piece.

This will generate a completely symettric (about the center) layout of quarters on the table. This means that whenever the opponent selects a free space to place a quarter in, the space opposite is guaranteed to be free as well. Since we are always guaranteed an open space, we will never lose with this strategy (and thus win when there are finally no more spaces for the opponent to use).

noodles aha:!

there is a pot of N noodles. (so there are 2N ends). a person randomly grabs two ends and merges them. the person keeps doing it, until there are no more noodles, (and only loops), left in the pot. what's the average number of loops in the pot?

thanks david

solution: noodles

solution: noodles

OK, all the answers so far just list formulae, without giving any explanation. I'll try to work it out out loud. (At this point I have no idea what the answer is.) Also, it's a math problem, so I'll assume we can ignore factors like how much the noodles stick together, how stiff they are, and so on. I'm sure I have no idea to take account of those factors.

Call the first end he picks up Noodle i. The second end he picks up is Noodle i*.

When he sticks the first two ends together, there are two possible outcomes:

(a) i* is the other end of noodle i, so he has created a loop.

(ii) i* is a different noodle from i, so he has created one long noodle out of two.

What are the odds of (a) happening? There are (2n-1) ends left in the bowl once he picks up the end of noodle i, and only 1 of them is the other end of the same noodle. Abstracting away from all physical details, let's say the odds of getting the other end of the same noodle are 1/(2n-1).

So the odds of (b) happening are 1-[1/(2n-1)], which is [(2n-1)-1]/(2n-1), i.e. (2n-2)/(2n-1).

If (a) happened, now we have a bowl with one loop in it, and n-1 other unlooped noodles. We add 1 to our count of loops, and repeat the problem for n-1.

If (b) happened, now we have a bowl with 0 loops in it, and n-1 other unlooped noodles. It's just that one of those noodles is especially long. We don't add anything to our count of loops; we just repeat the problem for n-1.

Now, when we get down to 1 unlooped noodle left in the bowl, the odds of (a) happening are 1.

The average # of loops will be: for each point where a loop could be formed, add 1 * the probability of a loop being formed then.

So we can write a function:

```
real average_number_of_loops (int n) {
  if (n == 1) {
   return 1
 } else {
   -- there is a 1/(2n-1) chance of getting 1 loop formed here
   -- and a (2n-2)/(2n-1) chance of getting 0 loops formed here
    -- and in either case we then have the same problem repeated for
    -- n-1 noodles
    -- so we should return ([(1/(2n-1))*1] + [(2/(2n-1))*0]) +
average_number_of_loops(n-1)
    -- or, simplifying...
    return (1/(2n-1)) + average_number_of_loops(n-1)
  }
```

Equivalently:

```
real average_number_of_loops (int n) {
  if (n == 0) {
   return 0
  } else {
   return (1/(2n-1)) + average_number_of_loops(n-1)
 }
```

So I guess I agree with the guy who wrote:

> Summation for i = 1 to N of 1 / 2i - 1. Sorry I can't figure out how to > resolve.

Except that you have to understand his "1 / 2i - 1" as being "1/ (2i - 1)." Which is no doubt what he intended. But now we have an explanation of why

heaven aha:!

a person dies, and arrives at the gate to heaven. there are three doors. one of them leads to heaven. another one leads to a 1-day stay at hell, and then back to the gate, and the other leads to a 2-day stay at hell, and then back to the gate. every time the person is back at the gate, the three doors are reshuffled. how long will it take the person to reach heaven?

this is a probability question - i.e. it is solvable and has nothing to do with religion, being sneaky, or how au dente the pasta might be ;-)

solution: heaven

solution: heaven

1/3 of the time, the door to heaven will be chosen, so 1/3 of the time it will take zero days. 1/3 of the time, the 1-day door is chosen; of those, the right door will be chosen the next day, so 1/9 trips take 1 day. Similarly, 1/9 will take two days (choosing the 2-day door, then the right door).

After that, the cases split again, and again, and again. I can't seem to make a nice infinite sum this way, so let's try again.

Suppose the average days spent is X. 1/3 of the cases are done in zero days as before. 1/3 of the cases are 1 day plus X. 1/3 are 2 + X. So:

$X = 1/3 * 0 + 1/3 * (1 + X) + 1/3 * (2 + X)$
$= 0 + 1/3 + X/3 + 2/3 + X/3$
$= 1 + 2X/3$

Therefore,

$X/3 = 1$
$X = 3$

On average, it takes three days to get to heaven. Two if the noodles are limp.
Took me one blind alley, and about five minutes. (heh heh

flipping coins aha:!

someone walks into your room and dumps a huge bag of quarters all over the floor. they spread them out so no quarters are on top of any other quarters. a robot then comes into the room and is programmed such that if it sees a head, it flips it to tails. if it sees a tail, it throws it in the air. the robot moves around randomly forever. will there be a convergence in distribution of heads vs. tails?

solution: flipping coins

solution: flipping coins

Hmmm. I made a little math trick out of it. If the 'bot finds a head, it flips. If the bot finds a tail, there's a fifty percent chance this will become a head, as well.

(P_h = #heads/#coins, P_t = #tails/#coins)

So, delta h = -P_h + .5 P_t

= -(1 - P_t) + .5 P_t

= 1.5 P_t -1

Which is zero when P_t is 2/3. It's important to remember that a flip to a tail results in no change to the number of tails -- this threw me off for a second.

pennies aha:!!

i challenge you to a game. we each get one penny and we flip them at the same time. (so on turn 1, we each flip our respective pennies - turn 2, we flip them again, and so on until someone wins). i am looking to get heads then tails. you are looking to get heads then heads. so if you flip heads on any flip and then heads on the next flip, you win. if i flip heads on any flip and then tails on the next flip, i win. (its not a speed race, we both flip at the same time, except i'm only concerned with what appears on my coin, and you are only concerned with whats on your coin). are the odds fair? (obviously not, otherwise this wouldn't be a question). who has the advantage and why?

thanks rob

solution: pennie

solution: pennies

(ummm,, I don't right too good. ;))

In a 3 turn game:

O needs HH, X needs HT

HHH O

HHT OX

HTH X

HTT X

THH O

THT X

TTH

TTT

X wins 4 times out of 8, O wins 3 times out of 8

Its because in the event X loses there is a 50% chance of starting out on another H where O has to start on a T 75% of the time. (or something like that...)

linked list aha:!!

how does one find a loop in a singly linked list in O(n) time using constant memory? you cannot modify the list in any way (and constant memory means the amount of memory required for the solution cannot be a function of n.)

solution: linked last

solution: linked list

I first figured this out when I was asked the question in a Microsoft interview, so there's verification that one question in the book was from a real interview. The best part of this problem is that I actually needed to write the code out for it a little while ago to detect a bug.

One way to detect a loop is to iterate over the list with 2 pointers at the same time where one is iterating at double speed. If the 2 pointers are ever equal after they iterate once and before they both reach an end, there's a loop.

Now for another puzzle.. I think the next question I had to answer was something along the lines of "OK, How do you remove a loop in a linked list with the same constraints?" The latter question definitely seems harder, but in the sequence of questions I'd already answered for the interviewer, the solution was pretty obvious. I'll leave that solution to someone else today.

Hm. I think I should pick up that book. I've always wondered where people got their juicy interview riddles.. I've always just repeated ones I've heard before or made up new ones along the same lines.

last ball aha:!

you have 20 blue balls and 14 red balls in a bag. you put your hand in and remove 2 at a time. if they're of the same color, you add a blue ball to the bag. if they're of different colors, you add a red ball to the bag. (assume you have a big supply of blue & red balls for this purpose. note: when you take the two balls out, you don't put them back in, so the number of balls in the bag keeps decreasing). what will be the color of the last ball left in the bag?

once you tackle that, what if there are 20 blue balls and 13 red balls to start with?

thanks to vijay varanasi

solution: last ball

solution: last ball

You always take off Red Balls two by two !

So if you start with 14 Red Balls, you cannot have one single Red ball at the end.

... so the last ball is blue.

But if you start with 13 Red Balls, as you take them off 2 by 2 (at one moment or the other you'll do it !) you will arrive at a moment when you have 1 red ball in the bag. But as you can only take off Red Balls 2 by 2 (Did I already say that ?!) you'll remove the last Blue Balls, one by one......

So the lastball will be red...

Oh, by the way, did I tell you that you take off Red Balls 2 by 2 ?! ;->

For tired people here is why you take off Red Balls 2 by 2 :
- If you take off 1 RED and 1 BLUE, in fact you will take off 1 BLUE
- If you take off 2 RED, in fact you will take off 2 RED (and add 1 BLUE)
- If you take off 2 BLUE, in fact you will take off 1 BLUE

coin rolls aha:!

every night, i dump all the change in my pocket into a big bucket.

when I buy things, i never hand over coins. always bills. so i accumulate a lot of coins. even if the purchase price is $1.01, and i have lots of coins in my pocket, i pay $2 and take the 99c in change. all the more coins to dump in my change bucket!

after about 10 years of this, i decide to roll all the coins into rolls. remember that a quarter roll is $10, a dime roll is $5, nickels $2, pennies 50 cents. so I go to the Banking Supply Store and buy empty paper rolls.

the Banking supply store, conveniently, sells assortment packs of coin rolls. each assortment pack contains W quarter rolls, X dime rolls, Y nickel rolls, and Z penny rolls.

the question: what is the optimum ratio of W to X to Y to Z to maximize the probability that I will use up the assortment packs at the same rate, e.g. without lots of leftover nickel tubes and stuff.

p.s. this problem should ideally be solved using Excel (but if you really like doing these things by hand, be my guest).

submitted by joel

paul brinkley makes these assumptions which are all good assumptions to make:
Assumption 1: The price of purchases made, modulo $1, is an even distribution from 0 cents to 99 cents.
Assumption 2: The cashier will always give you the least number of coins mathematically possible, and will always have enough of each type of coin to do this. So you'll never get 99 pennies as change for a $1.01 purchase, for example.
Assumption 3: Half dollars don't exist.

solution: coin rolls

solution: coin rolls

"Brute force" approach:

I guess I'll begin with a few assumptions:

Assumption 1: The price of purchases made, modulo $1, is an even distribution from 0 cents to 99 cents. Probably not true, but we can cover that later.

Assumption 2: The cashier will always give you the least number of coins mathematically possible, and will always have enough of each type of coin to do this. So you'll never get 99 pennies as change for a $1.01 purchase, for example.

Assumption 3: Half dollars don't exist.

Over the long haul, then, you'd get N sets of 0 cents, 1 cent, 2 cents, and so on up to 99 cents. So let's consider one of each. How many of each coin would you end up with?

Easy first: let's do quarters. 25-49 cents each gets you one. 50-74 each gets you two. 75-99 each gets you three. That's (1+2+3)*25, or 150 quarters.

Dimes next. 10-19 gets you one. 20-24, two. (You'd get a quarter instead for 25 and up.) 35-44, one. 45-49, two. 60-69, 85-94, one. 70-74, 95-99, two. So that's 4*(10+5*2), or 80 dimes.

Nickels. One for 5-9, 15-19, and that same pattern for +25, +50, +75, so it's 4*10 or 40 nickels. You'll never get two at a time, since a dime (at least) will do.

Pennies. Let's cut to the chase: 1,2,3, and 4 cents gets 10 pennies in all, and that pattern happens 20 times, so that's 200 pennies.

Let's double-check. 200 pennies, 40 nickels, 80 dimes, 150 quarters. That's 200*1 + 40*5 + 80*10 + 150*25 cents = 200+200+800+3750 = 4950 cents, which is the sum of all numbers from 0 to 99, so it checks.

15/8/4/20 would be the ratio then, IF coin rolls all hold the same number of coins, but they don't. Quarter rolls hold 40 coins, dime rolls hold 50, nickel rolls 40, penny rolls 50. So you need 5/4 as many quarter and nickel rolls. The final ratio is 75/32/20/80. Seems like a lot of quarters and pennies, except for the assumption that you'll tend to get them much more often than nickels and dimes as change.

The numbers change slightly when you figure in things like frequency of coins in circulation, the supply the cashier has at any one time, the actual distribution of change values, and the cashier's inclination to give you two dimes and a nickel instead of a quarter just because...

So what numbers do assortment packs really contain?

calendar cubes aha:!!!!

a man has two cubes on his desk. every day he arranges both cubes so that the front faces show the current day of the month. what numbers are on the faces of the cubes to allow this?

note the aha factor

thanks to kevin laws

solution: calendar cubes

solution: calendar cubes

Hmm. I don't know why this would warrant four aha's, but we'll see...

First, to show all possible days, we'd need one of each of the ten digits. We'd also need two 1s and two 2s to show 11 and 22. That's twelve numbers right there. Two cubes, twelve faces, so every face is used. Quite elegant.

We know each cube will need a 1 and a 2. Let's put the 3 on one of them. The 0 has to go on the other. We put 4, 5, and 6 on the 3-cube since we need to show 04 05 06.

But now where do 7, 8, and 9 go? The 0-cube needs them to be on the 3-cube, but it's full. I'm beginning to see why this gets four aha's.

Let's try this: clear both cubes, put the 0 on one of them. Now 1-9 have to go on the other cube to show 01-09. We can't put a 0 on the other cube, too, because that puts us over the 12-digit limit. It seems I have mathematical proof that this cannot be done! What am I missing? It's not like you can turn one of the other numbers sidewise to make another 0...

Heh.

You CAN make a 9 out of a 6, though. That frees up a digit. So you put 0, 1, and 2 on both cubes. Put 3 on one of them. 4 and 5 can go on it too. Put 6, 7, and 8 on the other. Now you can show 01-31 with no problem, and even 00 and 32 if you're feeling weird.

I'd've given it two, maybe three aha's. :)

boys and girls aha:!

in a country in which people only want boys, every family continues to have children until they have a boy. if they have a girl, they have another child. if they have a boy, they stop. what is the proportion of boys to girls in the country?

thanks to kevin again

solution: boys and girl

solution: boys and girls

Pretty simple. Half the couples have boys first, and stop. The rest have a girl. Of those, half have a boy second, and so on.

So suppose there are N couples. There will be N boys. There will be an "infinite" sum of girls equal to N/2 + N/4 + N/8 + ... As any college math student knows, this sum adds up to N. Therefore, the proportion of boys to girls will be pretty close to 1:1, with perhaps a few more boys than girls because the sum isn't actually infinite.

Btw, hope you enjoyed your fishing trip, Michael!

painfully easy aha:!

i flip a penny and a dime and hide the result from you. "one of the coins came up heads", i announce. what is the chance that the other coin also came up heads?

thanks tubby

solution: easy

solution: painfully easy

Assuming complete honesty on the part of the flipper, wouldn't the solution be 33%?

There are four possible scenarios:

HH
TH
HT
TT

Obviously the TT possibility can be discounted because it does not result in one of the two being heads.

This lees us with three possibilities, only one of which has the other coin also being heads.

Therefore one third.

I think.

I usually get these wrong.

chinese emperor aha:!

A chinese emperor had to choose a new adviser amongst 3 sages, all of them equally wise. He placed a problem to them: "To choose one of you, you'll play a simple and fair game: In this sack there are 3 white balls and 2 black balls. Each of you will be blindfolded and will pick one ball and place it on your head. After that, the blindfolds will be removed and each one in turn will try to guess the colour of the ball upon his head, by observation of the other picked balls. However, beware. You may pass your turn in guessing, but if you state a colour and fail, you're disqualified. This way I'll learn which one is the most intelligent amongst you" The sages talked briefly to each other and promptly refused: "Dear lord, it's of no use, since the game is not fair. The last one of us to guess in the first round will know the answer." and the sages promptly demonstrated this to the emperor, who was so amazed by their wits that he appointed all 3 has his advisers. Could you demonstrated it ? NOTE: If the emperor had any wits at all he would have named them all advisers in the first place... maybe spending reduction ? :)

nuggets aha:!

you can go to a fast food restaurant to buy chicken nuggets in 6-pack, 9-pack or 20-packs. is there such a number N, such that for all numbers bigger than or equal to N, you can buy that number of chicken nuggets?

solution: nuggets

solution: nuggest

Here's another way of looking at it:

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	21
22	23	24
25	26	27
28	29	30
31	32	33
34	35	36
37	38	39
40	41	42
43	44	45

(and so on, to infinity)

You can get any number in the right column except 3 by adding 6s and 9s. So cross out the entire right column except 3. You can also add 20 to any crossed-out number and cross that number out. So cross out everything in column two below and including 26. Finally, by the same logic you can add 20 to the crossed-out numbers in column 2 and thereby cross out everything in column one below and including 46.

The largest number that's left over is 43. Incidentally, cross out 20 and 40 and your map is complete.

orbs aha:!

you have two identical crystal orbs. you need to figure out how high an orb can fall from a 100 story building before it breaks. you know nothing about the toughness of the orbs: they may be very fragile and break when dropped from the first floor, or they may be so tough that dropping them from the 100th floor doesn't even harm them.

what is the largest number of orb-drops you would ever have to do in order to find the right floor? (i.e. what's the most efficient way you could drop the orbs to find your answer?)

you are allowed to break both orbs, provided that in doing so you uniquely identify the correct floor.

thanks to David Gerster

solution: or

solution: orbs

14.

Drop first orb from floors 14, 27, 39, 50, 60, 69, 77, 84, 90, 95, 99, 100... (ie move up 14 then 13, then 12 floors, etc) until it breaks (or doesn't at 100). Call the first floor at which it breaks n and the previous tested floor n'. Then try the intervening floors (n'+1 .. n'-1) with the other orb.

Worst case is if correct floor is 13,14,26,27, etc which require m drops with the first orb and 14-m drops with the second.

salary aha:!

three coworkers would like to know their average salary. how can they do it, without disclosing their own salaries?

solution: salarie

solution: salary

How about: Person A writes a number that is her salary plus a random amount (AS + AR) and hands it to B, without showing C. B then adds his salary plus a random amount (BS + BR) and passes to C (at each step, they write on a new paper and don't show the 3rd person). C adds CS + CR and passes to A. Now A subtracts her random number (AR), passes to B. B and C each subtract their random number and pass. After C is done, he shows the result and they divide by 3.

As has been noted already, there's no way to liar-proof the scheme.

It's also worth noting that once they know the average, any of the three knows the sum of the other 2 salaries.

world series aha:!

you have $10,000 dollars to place a double-or-nothing bet on the Yankees in the World Series (max 7 games, series is over once a team wins 4 games).

unfortunately, you can only bet on each individual game, not the series as a whole. how much should you bet on each game, so that, if the yanks win the whole series, you expect to get 20k, and if they lose, you expect 0?

basically, you know that there may be between 4 and 7 games, and you need to decide on a strategy so that whenever the series is over, your final outcome is the same as an overall double-or-nothing bet on the series.

solution: world series

solution: world series

this probably isn't the cleanest solution, but...

a dynamic-programming type solution is:

(1) Create a 5x5 matrix P.

So, P[i,j] holds your pile of money when the yanks have won i games and the mets have won j games.

initialize P[4,j] := 20 for j from 0 to 3 initialize P[i,4] := 0 for i from 0 to 3

fill P in bottom-right to top left by averaging bottom and right adjacent cells:

P[i,j] := (P[i+1,j]+P[i,j+1]) / 2

(2) Make another 5x5 matrix, B, which represets your bet at any-time.

So, B[i,j] represents your bet when the yanks have won i games and the Mets j games.

fill this top-left to bottom right by:

B[i,j] = P[i+1,j] - P[i,j]

(3) Look in matrix B for your bet at any time.

The final matricies are:

Pile-Matrix

	0.00	1.00	2.00	3.00	4.00
0	10.00	6.88	3.75	1.25	0.00
1	13.13	10.00	6.25	2.50	0.00
2	16.25	13.75	10.00	5.00	0.00
3	18.75	17.50	15.00	10.00	0.00
4	20.00	20.00	20.00	20.00	

Bet-Matrix

	0	1	2	3	4
0	3.13	3.13	2.50	1.25	
1	3.13	3.75	3.75	2.50	
2	2.50	3.75	5.00	5.00	
3	1.25	2.50	5.00	10.00	
4					

00 factorial aha:!

how many **trailing** zeroes are there in 100! (100 factorial)?

thanks to dallow - creator of Â© atomize

solution: 100!

solution: 100 factorial

One per factor of 10, and one per factor of 5 (there are more than enough 2's to pair with the 5's), plus one per factor of ten squared (one occurrence) and one per factor of 5 squared (three occurrences).

So if I'm counting correctly, that'd be 10 + 10 + 1 + 3== 24 zeroes.

Assuming the question meant *trailing* zeroes. It'd be much harder to also count the intermingled zero digits in the entire expansion.

oil mogul aha:!

you are an oil mogul considering the purchase of drilling rights to an as yet unexplored tract of land.

the well's expected value to its current owners is uniformly distributed over [$1..$100]. (i.e., a 1% chance it's worth each value b/w $1..$100, inclusive).

bcause you have greater economies of scale than the current owners, the well will actually be worth 50% more to you than to them (but they don't know this).

the catch: although you must bid on the well before drilling starts (and hence, before the actual yield of the well is known), the current owner can wait until *after* the well's actual value is ascertained before accepting your bid or not.

what should you bid?

thanks to tom and jay

solution: oil

solution: oil mogul

This problem amounts to properly defining the expected value of the well to you.

The following equation does it:

(1%) * [(1.5 - 1)] +

(1%) * [(3 - 2) + (1.50 - 2)] +

(1%) * [(4.5 - 3) + (3 - 3) + (1.5 - 3)] +

. . .

(1%) * [(150-100) + ... + (3-100) + (1.5-100)]

Each line represents your expected value from a bid of 1$, 2$, ..., 100$, respectively.

eg, consider line 2 above. if you bid $2...

With 98% probability you won't win the contract, so your profit is 0. With 1% probability, you will win something worth (150%*1) = 1.5, for which you paid 2$
With 1% probability, you will something worth worth (150%*2) = 3, for which you paid 2$

So, your goal is to maximize the following function of x, where x is your bid.

f(x) = 1% * Sum_{i = 1 to floor(x)}{ x - 1.5*i }

There's no benefit to non-integer bets, so re-write the maximization function as :

ARGMAX(k) {1% * Sum_{i = 1 to k}{1.5*i - k}}

(=) ARGMAX(k) {Sum_{i=1 to k}{1.5*i - k}} /* 1% isn't a function of k or i, so toss it */

(=) ARGMAX(k) {Sum_{i=1 to k}{1.5*i} - Sum_{i=1 to k}{k}} /* Split the summation */

(=) ARGMAX(k) {(0.75)(K)(K+1) - K^2 }} /* Closed form for summations */

(=) ARGMAX(k) {(0.75)(k)-(0.25)(K^2)}} /* Algebra */

And that function is maximized at k = 1 and k = 2.

When choosing b/w $1 and $2, you should bid $1 because of time-value, reinvestment risk, etc, of the extra dollar.

(ie, if you don't have to spend the extra $$ now, don't)

That's my solution.

vienna aha:!!

it's the middle ages, you're travelling across europe and you want to find the way to vienna. you come to a crossroads, now there are two ways to go. at the crossroads stand a knight and a knave. the knight answers every question truthfully. the knave answers every question falsely. you don't know which guy is which. how can you figure out which road leads to Vienna by only asking one question?

thanks dr. james and reuven

solution: vienna

solution: vienna

Many people try "what would the other man say if I asked him which way to Vienna?" But for that to work, the knight would have to know that the other man is a knave, and the knave would have to know that the other man is a knight; it's not clear that either of these is a given.

To cope with that, you can ask "assuming the other man has the opposite predilection regarding truth-telling from what you do, what would he say if I asked him which way to Vienna?"

More interestingly, you can ask "if you were to ask yourself which way to Vienna, what would you say?" But before risking that question, I'd want a few more details about how the knave is wired up. What does it mean, to lie to oneself? Is such a thing even possible, regardless of what psychoanalysts might claim?

Furthermore, travelling through Europe in the middle ages, it's a bit of a rash assumption that people you meet at a crossroad would understand English. If you only have 1 question, you have to frame a question that is meaningful in all locally spoken languages (like writing a program that is both legal Fortran and legal C). Your question doesn't have to mean exactly the same thing in each language, as long as each possible answer carries in itself an indication of which language it is in.

Or you could try the universal language of mime. Say "Vienna" and open your arms wide with a questioning look. But how would you mime "if you were to ask the other man ..." (or "if you were to ask yourself ...") ?

duel aha:!!

you find yourself in a duel with two other gunmen. you shoot with 33% accuracy, and the other two shoot with 100% and 50% accuracy, respectively. the rules of the duel are one shot per-person per-round. the shooting order is from worst shooter to best shooter, so you go first, the 50% guy goes second, and the 100% guy goes third.

where or who should you shoot at in round 1?

thanks to tom

solution: duel

solution: duel

You have 3 options to consider:

(1) Shoot at 50% guy. Death comes with: 72% likilhood (2) Shoot at the 100% guy. Death comes with: 63.64% likihood (3) Shoot into the air (purposely miss). Death comew with 58.33% likihood.

So, you shoot into the air and hope for the best.

Calculations:

The p in the probabilities below represents your probability of dying at some point if you MISS either guy in the first round. It'll be filled in later.

(1) if you shoot at the 50% guy and get him, you're guaranteed to get shot the next round.

prob of dying at some point by aiming at the 50% guy:

(=) (33.3%)(100%)+(66.67%)*(p)

(=) (33.3%)+(66.67%)(p)

(2) if you shoot at the 100% guy and get him, you're left with this geometric sum of probability of getting shot by the 50% guy at some point in the future...it converges.

(=) [(50%)] +

[(50%)(66.67%)(50%)] +

[(50%)(66.67%)(50%)(66.67%)(50%)] + ...

(=) 50% * { 1 + (1/3) + (1/3)^2 + ... }

(=) 50% * {3/2)

(=) 75%

so, your prob of getting shot if you shoot at the 100% guy is:

(=) (33.3%)(75%)+(66.67%)(p)

(=) (24.75%) + (66.67%)(p)

still crummy, but it dominates the alternative, for positive p.

(3) What happens if you miss? ie, what's (p)?

if you miss, the second guy has a choice to make...does he shoot at you, or the other guy? his options:

(1) Shoot you: if he gets you, he's guaranteed to die on the next shot.

if he misses, he has q chance of dying (which we'll get later).

(=) (50%)(100%)+(50%)(q)

(=) (50%)+(0.5)(q)

(2) Shoot at the 100% dude:

If he gets the 100% guy, there some infinite sum representing his probability of getting shot by you:

[(33.33%)] +

[(66.67%)(50%)(33.33%)] +

[(66.67%)(50%)(66.67%)(50%)(33.33%)] + ...

(=) 33.33% * [1 + (1/3) + (1/3)^2 + ...]

(=) 33.33% * [(3/2)]

(=) (50.0%)

So, he chances of getting shot if he shoots at the 100% guy are:

(=) (50%)*(50%) + (50%)(q)

(=) (25%)+(.5)q

No matter what q is, he'd prefer shooting the 100% guy to shooting you.

Now, what happens if *he* misses (the 100%) guy? ie, what's q? if he misses, then the 100% guy has to make a decision:

(1) Shoot you:

he's guaranteed to get you, so his chances of dying are just 50%. (game ends after the next round...the 50% gets only one shot)

prob of dying: 50%

(2) Shoot the 50% guy:

by the same logic, he'd have a 33.33% chance of death (getting shot by you).

So, he prefers to shoot the 50% guy.

So, q is 100% (ie, if the 50% guy misses, the 100% guy shoots him immediately). From that, we know the 50% guy's optimal move, if everyone's around on his first shot. He'd prefer shooting at the 100% guy to shooting at you or purposely missing.

Now, we need to get p, which is our probability of dying if we purposely miss either guy (shoot into the air).

We know the 50% guy shoots at the 100% guy if we miss.

(1) With 50% probability, the 50% guy kills the 100% guy, resulting in a shootout b/w us and the 50% guy.

(=) (50%) * [(66.67%)*(50%) + (66.67%)*(50%)*(66.67%)(50%) + ...

(=) (50%) * [(1/3) + (1/3)^2 + ...]

(=) (50%) * [0.5]

(=) (25%)

(2) He misses him. Then, we get one shot at the 100% guy (after the 100% guy shoots the 50% guy).

Our chances of death are:

(50%)*(66.67%) = (33.33%)

So, p is (25%)+(33.33%) = 58.3%

So, again, our three choices are

(1) Shoot at 50% guy. Death comes with:

(=) (33.3%)+(66.67%)(p)

(=) 72% likilhood

(2) Shoot at the 100% guy. Death comes with:

(=) (24.75%) + (66.67%)(p)

(=) 63.64% likihood

(3) Shoot into the air (purposely miss).

(=) p

(=) 58.33% likelihood.

hen aha:!!!

if a hen and a half lay an egg and a half in a day and a half, how many hens does it take to lay six eggs in six days?

thanks carl

solution: h

solution: hen

if 1.5 hens lay 1.5 eggs in 1.5 days (or 36 hours) then: 1 hen lays 1 egg in 1,5 days or 4 eggs in six days thus 1.5 hens lay 6 eggs in 6 days

mensa aha:!

if "24 H in a D" means "24 hours in a day", what does "26 L of the A" mean? what about the other 32 puzzles here on my mensa forward reproduction page?

see the others at mensa

thanks matt

i received this as an .xls attachment in the mail. i would just post the excel file, but i don't know if it has a virus or not, so i'm just going to copy the text here:

0	24 H in a D—24 hours in a day
1	26 L of the A – 26 letters of the alphabets
2	7 D of the W—7 days of the week
3	7 W of the W – 7 wonders of the world
4	12 S of the Z – 12 signs of the zodiac
5	66 B of the B—66 books of the bible
6	52 C in a P (WJs)
7	13 S in the USF—13 stripes of the US flag
8	18 H on a G C—18 holes of the golf course
9	39 B of the O T—39 books of the old testament(guess)
10	5 T on a F – 5 tyres in the car including Trunk(guess)

11 90 D in a R A – 90 degrees in a right angle
12 3 B M (S H T R) – 3 blind mice(see how they run)
13 32 is the T in D F at which W F—32 is the temp in degrees farenheidt at which water freezes
14 15 P in a R T
15 3 W on a T
16 100 C in a R
17 11 P in a F (S) T—11 players in a football (soccer) team
18 12 M in a Y – 12 months in a year
19 13=UFS
20 8 T on a O – think 8 triangles on an octagon.
21 29 D in F in a L Y—29 days in February in a leap year
22 27 B in the N T
23 365 D in a Y—365 days in a year
24 13 L in a B D--
25 52 W in a Y—52 weeks in an year
26 9 L of a C
27 60 M in a H -- 60 min in an hour
28 23 P of C in the H B
29 64 S on a C B—64 squares in a chess board
30 9 P in S S – 9 planets in the solar system
31 6 B to an O in C
32 1000 Y in a M --1000 years in a millenium
33 15 M on a D M C—15 men on a dead mans chest

you have to figure out what they all mean. for example, the first one is "24 hours in a day".
i'll post the answers later (post your answers on the discussion board if you feel the need).

---

box o' numbers aha:!

arrange the numbers 1 to 8 in the grid below such that adjacent numbers are not in adjacent boxes (horizontally, vertically, or diagonally).

```
    ___
   | 1 |
 =============
| 6 | 4 | 3 |
 =============
| 2 | 7 | 5 |
 =============
   | 8 |
   =====
```

the arrangement above, for example, is wrong because 3 & 4, 4 & 5, 6 & 7, and 7 & 8 are adjacent.
thanks to VJ for this one

solution: box 'o numbers

solution: box o' numbers

The key (as I see it) is putting the 1 & 8 in the centre spots - which is required, because those spots both border all but one of the other spots, and 1 & 8 are the only numbers that are only adjacent to one number.

From there, the 2 & 7 are forced, then you have your choice of the next number, which then forces the rest. Really, though there are only two valid solutions, and they are mirror images of each other.

even harder coin problem aha:!

some of you may have easily solved the pill weighing problem posed here. if so you are going to love this problem.

it is similar but much more difficult.

from my buddy tom:

> ok, here's a tough one (i thought). there are no "aha!" tricks - it
> requires straightforward deductive-reasoning.
>
> you have 12 coins. one of them is counterfeit. all the good coins weigh
> the same, while the counterfeit one weights either more or less than a
> good coin.
>
> your task is to find the counterfeit coin using a balance-scale in 3
> weighs. moreover, you want to say whether the coin weighs more or less
> than is should and, and this is the real kicker, your weighs must be
> non-adaptive.
>
> that is, your choice of what to put on the balance for your
> second weigh cannot depend on the outcome of the first weigh and your
> decision about what to weigh for round 3 cannot depend on what happened on
> either your first or second weigh.
>
> for example, you can't say something like "take coin #1 and coin #2 and
> weigh them. if they balance, then take coins 3,4,5 and weight them
> against 6,7,8...if 1 and 2 don't balance, then weigh #1 vs #12..." you
> have to say something like:
>
> round #1: do this
> round #2: do this
> round #3: do this
>
> if the results are left tilt, balanced, and left tilt, respectively, then
> coin #11 is heavier than it should be.
>
> this problem is solveable...it took me about 1-2 hours of working on it to
> get it. i think even finding the counterfeit using an adaptive solution
> is tough. then non-adaptive constraint makes it quite hard and having to
> find whether it's heavier and lighter is cruel and unusual riddling ;-)

>
> have fun...

technical riddles aha:!!

You have an empty room, and a group of people waiting outside the room. At each step, you may either get one person into the room, or get one out. Can you make subsequent steps, so that every possible combination of people is achieved exactly once?

x implies y aha:!!

Part I

(you can't use paper, you have to figure it out in your head)

i have a black triangle, a white triangle, a black circle and a white circle. if i gave you a shape (triangle or circle) and a color (black or white), the "frobby" items would be those that had either the shape or the color, but not both. that is, in order to be frobby, the item must be of the specified color OR the specified shape, but not both the specified shape AND the specified color. i'm thinking of a shape and a color in my head and i tell you that the white triangle is frobby. can you tell me the "frobbiness" of the other items?

Part II

there are four cards which have a letter on one side and a number on the other side. i lay them out and the cards appear as 2 5 F E. the rule is that a card with an odd number on one side must have a vowel on the other. what is the minimum number of cards you should turn over to prove the rule is true (and which cards would they be)?

I can't remember if I saw this on the discussion board or somewhere else...

Create an XOR gate using only NAND gates. (BTW, mostly all circuit problems assume you have two inputs plus 0 and 1 also as inputs).

smart cookie aha:?

did you ever wonder how they make those pillsbury cookie dough rolls with the intricate faces inside them? scroll about halfway down the page here and notice the intricate design they have somehow injected into their cookie rolls? if you examine the roll closely there is no seam between the normal dough and the colored shape, but somehow they get that inside the roll. i emailed them asking them how they do it and they told me it was "doughboy magic".

maybe the solution

solution: smart cookie

I don't know if this is right or not, but it sounds pretty darn close to the best idea I've heard yet.

From Brad Siemssen:

> "I assume these are made the same way designs are put in the center of taffy candies.
>
> The designs are made by taking advantage of the fact that the dough is soft and pliable.
>
> They take the dough and make the design at a large diameter and thickness. The diameter could a foot or more. At this scale the design does not have to be very precise. Extrusion methods like the playdough presses would work.
>
> They then take the giant dough cylinder with the rough design and then roll it into a "snake" just like kids do with balls of playdoh. As the "snake" gets longer the design in the center gets uniformly smaller. The "snake" is then cut into pieces and put in packaging.
>
> To get a more precise designs simply requires increasing the starting diameter. The engineering difficulty then lies in rolling the snake out.
>
> An alternative to rolling would be to use a press to extrude the dough using a funnel shaped press. The engineering difficulty with a press would be to shape the press, and exert uniform pressure so the dough coming out is a shrunken version of the original.

crazy guy on the airplane aha:!

"a line of 100 airline passengers is waiting to board a plane. they each hold a ticket to one of the 100 seats on that flight. (for convenience, let's say that the nth passenger in line has a ticket for the seat number n.)

unfortunately, the first person in line is crazy, and will ignore the seat number on their ticket, picking a random seat to occupy. all of the other passengers are quite normal, and will go to their proper seat unless it is already occupied. if it is occupied, they will then find a free seat to sit in, at random.

what is the probability that the last (100th) person to board the plane will sit in their proper seat (#100)?"

thanks to levik for the latest puzzle

wanna play? aha:!

i offer to play a card game with you using a normal deck of 52 cards. the rules of the game are that we will turn over two cards at a time. if the cards are both black, they go into my pile. if they are both red, they go into your pile. if there is one red and one black, they go into the discard pile. we repeat the two card flipping until we've gone through all 52 cards. whoever has more cards in their pile at the end wins. i win if there is a tie. if you win, i pay you a dollar. how much would you pay to play this game?

solution

solution: wanna play?

I wouldn't give you a penny for that game. Here's why.

Let's say all the color pairs are matched. That means 13 red pairs, 13 black pairs, you win.

Now let's say there is 12 red pairs in the deck, we pick them all from the top of the deck. With the 28 remaining card, there will always be exactly 12 black pairs and two mixed pairs. Why? Two red cards in the stack, they can't be in a pair, otherwise you would get a total of 13 pairs, and I would too since all the black cards would be together. So the two red cards are matched with one black card each, leaving 24 black cards together, I get the same number of pairs as you do and you win.

Same math with 11 red pairs. That means four red cards in mixed pairs, hence 22 black cards left together, or 11 pairs.

No matter how mixed up the cards are, it's a tie, and you win. Hence, I wouldn't give you a penny.

chameleons aha:!

"at one point, a remote island's population of chameleons was divided as follows:

- **13** red chameleons
- **15** green chameleons
- **17** blue chameleons

each time two different colored chameleons would meet, they would change their color to the third one. (*i.e.. If green meets red, they both change their color to blue.*) is it ever possible for all chameleons to become the same color? why or why not?"
another one from levik

railroad bridge aha:!!

i got some new brain teasers and more classic programming puzzles for everyone!

a man needs to go through a train tunnel. he starts through the tunnel and when he gets 1/4 the way through the tunnel, he hears the train whistle behind him. you don't know how far away the train is, or how fast it is going, (or how fast he is going). all you know is that

1. if the man turns around and runs back the way he came, he will just barely make it out of the tunnel alive before the train hits him.
2. if the man keeps running through the tunnel, he will also just barely make it out of the tunnel alive before the train hits him.

assume the man runs the same speed whether he goes back to the start or continues on through the tunnel. also assume that he accelerates to his top speed instantaneously. assume the train misses him by an infintisimal amount and all those other reasonable assumptions that go along with puzzles like this so that some wanker doesn't say the problem isn't well defined.
how fast is the train going compared to the man?

cars on the road aha:

if the probability of observing a car in 20 minutes on a highway is 609/625, what is the probability of observing a car in 5 minutes (assuming constant default probability)?

thanks to joel w. for this one

haha. MS questions again!

Using two cubes (12 sides) make a desk calendar that can cover all of
the days of the month (01-31).

Generate a 9 digit number using the number 1-9, without using any number
more than once. this number must also be divisible such that if you take
the left most digit the number is divisible by 1, if you take the two
left most digits the number is divisible by 2, if you take the three left
most digits the number is divisible by 3, etc., etc. What is the number?
There was a sheriff in a town who caught three outlaws. He said he was
going to give then all a chance to go free. All they had to do is figure
out what color hat they were wearing. There sheriff had 5 hats 3 black and
2 white. The first guy guessed and was wrong so he was put in jail. The

second guy also guessed and was also put in jail. finally, a third blind man guessed and he guessed correctly. How did he know?

Design the ideal alarm clock. Explain it's features.

Write all the string functions in as few lines as possible.

```
/* copy string ct to string s, including '\0', return s */
void * strcpy(char *s, const char *ct)
{

}

/* copy at most n characters of string ct to s; return s.
Pad w/ '\0' if ct has fewer than n characters */
void * strncpy(char *s, const char *ct, size_t n)
{

}

/* concatenate string ct to the end of string s; return s */
void *strcat(char *s, const char *ct)
{

}

/* concatenate at most n characters of string ct to string s,
terminate s with '\0'; return s */
void *strncat(char *s, const char *ct, size_t n)
{

}
```

All those are basic interview question in C/C++/Windows.

Alex

===

To: zephyr@howdyneighbor.com
Subject: MS Interview Questions.

Hi,

Your web-page about the Microsoft Interview was very much useful to me. Thank u very much for that.

I had a telephonic Interview last week with Microsoft. I didnt expect such a simple interview from Microsoft. In spite of the Interview being simple, I made few blunders. But, to my surprise, they informed me coupla days ago that they wanted to interview again on this week. I have no idea is it going to be the second level of the interview or else?

If you have any idea regarding this and the kind of questions that'd be asked, will u please email me at mohan24@hotmail.com ?

I am herewith sending the questions asked to me. Hope u'll find it useful!!!

Please respond this mail ASAP.

Bye
XXXXXXXX

The Interview was 3 phased in C, C++ and Windows Prog. Some of my answers may not be convincing to u. If so, please reply me what do u feel as the correct answer.

Q : What is Operator, Operand, Expression, Statement in 'C' ?

A : Operators are symbols which take one or two or three operands or expressions and performs arithmetic or logical computations.
There are three type of operators in 'C', they are: unary operators binary operators and trenary operators.

Operands are variables or expressions which are used as parameters for
the operators to evaluate the expression.

Expressions can be a combination of operands and operator in a legal structure.

Statements are combination of expressions and keywords.

Q : Given
int a=1, b=2, c=3; what does the following expressions evaluate to?

A : a < b will evaluate to TRUE

a < b && c < b will evaluate to FALSE

a <= c or ? will evaluate to TRUE

(a+=3) < c will evaluate to FALSE

Q : Given int n, i=10, j=20, x=3, y = 100;
what is the value of n and y at the end of each of the foll. expression.

A : a) n = (i > j) && (x < ++y);

after the expr. the values of n and y will be

n = 0 and y = 100. y will not be incremented since the first expression
itself evaluates to FALSE. A FALSE anded with any expression is FALSE.
So, the second expression will not be evaluated

b) n = (j - i) && (x < y++);

the result n = 1 and y = 101

c) n = (i < j) || (y+=i);

the result n = 1 and y = 100

d) I am afraid I did not note down the 4th expression.

Q : int x = 5;
int y = 7;
what is the value of x and y after the expression y+=x++;

A : x = 6 and y = 12

Q : if the expression is y+=++x;, what is the the value of y?

A : y will be 13 in this case

Q : How do you multiply a variable by 16 without using the multiplication
operator '*'.

A : We can use bitwise shift operator to achieve the result. Supposing x is
a int variable to be multiplied by 16. We can have the expression

x <<= 4; to achive the result.

Q : What if you have to multiply by 15?

A : We can bitwise left shift the variable by 4 times and subtract 16 from the resul.

Q : Convert 27 to Hexadecimal notation

A : 1B

Q : Explain the following declaration : long foo(char);

A : it's a function declaration to notify the compiler that foo is a function which takes a char parameter and returns a long variable.

Q : write a function declaration for a function which returns int * and takes float, int and char * as parameters

A : int * func(float, int, char*)

Q : what is the function prototype of printf() in 'C' and what does it return

A : int printf(char *, ...);
it returns the no. of bytes printed. Here ... means it takes variable no. of parameters

Q : int (*a) [10];
a++;
What is the value of a after this?

A : a will be incremented by 20, assuming the size of int is 2.

Q :
```
main()
{
   int *ptr;
   *ptr=10;
}
```
what is wrong with this?

A : it will raise an error. Because, ptr does not point to any memory location so assigning a value to it will result in error.

Q : typedefine a function pointer which takes a int and float as parameter
and returns a float *.

A : typedef float* (*fp) (int, float);

Q : which of the foll. expression will affect the values of pa or pb

A : a) pb - pa;
does not affect pa or pb

b) ++pa;
affects pa

c) pa + i
does not affect pa or pb

d) pa < pb
does not affect pa or pb

Q : Is the following operation illegal? assume pa and pb as pointers

A : a) pa * pb
illegal

b) pa + pb
illegal

c) pa - pb
legal

d) pa << pb
illegal

Then various questions were asked in C++ and Windows programming.

I did not take a note of it.

some of them include

what is the advantage of OOP?
what is polymorphism?
the diff. among struct, class and union?
recursive function mechanism

virtual classes and virtual functions

how does Windows prog. differs from DOS prog.
what is the advantage of Windows
what does winmain and wndproc do?
how does user input are trapped in Windows?
Windows Messaging mechanism
.... ETC....

The following is the questionire which they sent me to be used while interviewing. But, no questions were asked from this.

Q:

```
#include <stdio.h>

void main(void)
{
      char array1[6] = {"squid"};
      char array2[6] = {"llama"};

      char *ptr1 = array1;
      char *ptr2 = array2;

      printf("*ptr1 = %s\t*ptr2 = %s\n\n", ptr1, ptr2);

      swap(&ptr1, &ptr2);

      printf("*ptr1 = %s\t*ptr2 = %s\n\n", ptr1, ptr2);
}
```

A : The question is not given for this problem. Assuming, I am asked to find out
the output of the program. The first printf would print

```
*ptr1 = squid   *ptr2 = llama
```

The second printf functions output cannot be determined unless we know
what is done inside the swap(). The swap function takes a pointer to pointer
parameter

Q : What is the output of the following program? How can the code be changed to give the expected output?

```
        #include <stdio.h>

        #define ONE          1
        #define TWO          2
        #define NEG1    ONE - TWO
        #define THREE   TWO - NEG1

        void main( void )
        {
            printf( "ONE = %d\n", ONE );
            printf( "TWO = %d\n", TWO );
            printf( "THREE = %d\n", THREE );
        }
```

A : The output is

```
            ONE = 1
            TWO =  2
            THREE = -1
```

We have to change the the third macro to

```
#define THREE   TWO - (NEG1)
```

to get the desired output

Q :

```
#include <stdio.h>
#include <stdlib.h>

void main(int, char *[]);
void foo(int);

void main(int argc, char *argv[])
{
    int x;

    x = atoi(argv[1]);
    foo(x);
}

void foo(int x)
{
    int count=0;
```

```
        while (x)
                x &= (x-1), count++;
        printf ("count = %1d\n", count);
}
```

Command-line to execute program:

```
> test 5
```

A : the output is

```
        count = 2
```

Q : Is there anything wrong with the following code? If so, how would
you
fix it?

```
class X
{
protected:
  int a;
public:
  X(int n) { a = n; }
};

void somefunction()
{
  X* pX = new X[12];
}
```

A : Yes, it does not have the default constructor. I can add the
constructor

```
    X::X() {}
```

to the class to fix it

Q : Given:

```
class String
{
        int length;
        char * buffer;
public:
        String();
        String(const char *s);
        ~String();
};
```

Write the copy constructor for this class.

```
A : String :: String ( const String &s)
    {
        length = s.length;
        buffer = new char[length];
        strcpy(buffer, s.buffer);
     }
```

Q : What is the output of the printf statement in the following code snippet
if it is compiled as part of a C program? What if it's part of a C++ program?

```
void f()
{
        int x;
        x = 5;
        g(x);
        printf("x = %d\n", x);
}
```

A : if it's a 'C' program, the printf function will always print x = 5

if it's a 'C++' program, there's a possibility that the g(x) function
takes x as a reference and change the value of x. So, the x value may
differ in the printf function

Q : Read and explain the following code:

```
class X
{
        int n;
        char *pChar;
public:
        X();
        X(int a = 0);
};

void f()
{
        X x;
        x.n = 4;
}
```

A : it has 2 errors. The constructors will lead to ambiguity. And x.n is not accessible because n is private member data.

Q : Implement a constructor for this class which initializes the member variable n with the value passed as a.

A : x(int a){ n = a;}

Q : How would you initialize n with a if n were declared const?

A : x(int a) : n(a) { }

Q : If n is a static, non-const member of class X, how would you initialize it?

A : int X :: n = 10;

Q : Assuming that the constructor also allocates memory for pChar, what problems might arise from using this class? How would you eliminate these problems?

A : When the object is destroyed, the memory allocated for that does not get released. So, we can write a destructor as

```
X :: ~X() { delete[] pChar;}
```

Q : What is the output from this program?

```
#include <iostream.h>

class Base
{
 public:
   Base() { vf(); }
   ~Base() { vf(); }
   virtual void vf() { cout << "Base::vf()" << endl; }
};

class Derived : public Base
{
 public:
   Derived() { };
   ~Derived() { };
   virtual void vf() { cout << "Derived::vf()" << endl; }
};

void main()
{
 Base* pb = new Derived;

 pb->vf();

 delete pb;
}
```

A : the output is

```
Base::vf()
Derived::vf()
Base::vf()
```

Q : What problems if any do you see in the following code fragment?
(Assume
that X is some user-defined class and that E is some user-defined class

used to encapsulate the details of error conditions. GetChunk is some utility function which should normally return a non-NULL pointer.)

```
void f()
{
  try
  {
    g();
  }
  catch(E e)
  {
              cerr << e.text << endl;
  }
}

void g()
{
  X* pX = new X;
  h(pX);
  delete pX;
}

void h(X* pX)
{
  // ...

        if( GetChunk() == NULL )
              throw E("No chunk available");
  // ...
}
```

A : No problem is perceived

Q : What is the output of the following program?

```
#include <iostream.h>

void main()
{
        int i = 5;
        int j = 4;
        int& r1 = i;
        int& r2 = j;
        int* p1 = &r1;
        int* p2 = &r2;
        *p1 = 3;
        *p2 = 2;
        cout << "i = " << i << endl;
        cout << "j = " << j << endl;
        r1 = r2;
        p2 = &r1;
        *p2 = 1;
        cout << "i = " << i << endl;
        cout << "j = " << j << endl;
}
```

A : the output is

```
i = 3
j = 2
i = 1
j = 2
```

Q : What's the output of the following program?

```
#include <iostream.h>

class X
{
public:
  X(char* s) { cout << s << endl; }
};

class Y : public X
{
  X a;
  X b;
public:
  Y() : b("b"), a("a"), X("base") {}
};

void main()
{
  Y y;
}
```

A : the output is

base
a
b

Q : Will this work?

```
void X::SwitchMe(X* anotherX)
{
  this = anotherX;
}
```

A : it should not work.

Q : CODEGEN.C

```
        int globalint;

        void function(int i)
        {
              return i+1;
        }

        void main()
        {
              globalint = function(1);
        }
```

A : there will be a compile time error for this program. Because the
return
value of function is void which cannot be assigned to globalint.

GENERIC.C

```
#include "windows.h"
#include "generic.h"

HANDLE hInst;

int PASCAL WinMain(hInstance, hPrevInstance, lpCmdLine, nCmdShow)
HANDLE hInstance;
HANDLE hPrevInstance;
```

```
LPSTR lpCmdLine;
int nCmdShow;
{
   MSG msg;

   if (!hPrevInstance)
      if (!InitApplication(hInstance))
         return (FALSE);

if (!InitInstance(hInstance, nCmdShow))
      return (FALSE);

while (GetMessage(&msg,
         NULL,
         NULL,
         NULL))
      {
      TranslateMessage(&msg);
      DispatchMessage(&msg);
   }
   return (msg.wParam);
}


BOOL InitApplication(hInstance)
HANDLE hInstance;
{
   WNDCLASS  wc;


   wc.style = NULL;
   wc.lpfnWndProc = MainWndProc;
   wc.cbClsExtra = 0;
   wc.cbWndExtra = 0;
   wc.hInstance = hInstance;
   wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
   wc.hCursor = LoadCursor(NULL, IDC_ARROW);
   wc.hbrBackground = GetStockObject(WHITE_BRUSH);
   wc.lpszMenuName =  "GenericMenu";
   wc.lpszClassName = "GenericWClass";

   return (RegisterClass(&wc));

}
```

```
BOOL InitInstance(hInstance, nCmdShow)
   HANDLE       hInstance;
   int          nCmdShow;
{
   HWND         hWnd;

   hInst = hInstance;

   hWnd = CreateWindow(
      "GenericWClass",
      "Generic Sample Application",
      WS_OVERLAPPEDWINDOW,
      CW_USEDEFAULT,
      CW_USEDEFAULT,
      CW_USEDEFAULT,
      CW_USEDEFAULT,
      NULL,
      NULL,
      hInstance,
      NULL
   );


   if (!hWnd)
      return (FALSE);


   ShowWindow(hWnd, nCmdShow);
   UpdateWindow(hWnd);
   return (TRUE);

}


long FAR PASCAL MainWndProc(hWnd, message, wParam, lParam)
HWND hWnd;
UINT message;
WPARAM wParam;
LPARAM lParam;
{
   FARPROC lpProcAbout;

   switch (message) {
      case WM_COMMAND:
            if (wParam == IDM_ABOUT) {
                  lpProcAbout = MakeProcInstance(About, hInst);

                  DialogBox(hInst,
                    "AboutBox",
                    hWnd,
                    lpProcAbout);

                  FreeProcInstance(lpProcAbout);
                  break;
            }
            else
           return(DefWindowProc(hWnd,message,wParam,lParam));

        case WM_DESTROY:
            PostQuitMessage(0);
            break;

        default:
            return (DefWindowProc(hWnd,message, wParam, lParam));
    }
    return (NULL);
}



BOOL FAR PASCAL About(hDlg, message, wParam, lParam)
HWND hDlg;
unsigned message;
WORD wParam;
LONG lParam;
{
    switch (message) {
        case WM_INITDIALOG:
            return (TRUE);

        case WM_COMMAND:
            if (wParam == IDOK
                || wParam == IDCANCEL) {
                EndDialog(hDlg, TRUE);
                return (TRUE);
            }
            break;
    }
    return (FALSE);
```

```
}
```

--
※ 来源:．The unknown SPACE bbs.mit.edu．[FROM: 128.174.97.78]

[本讨论区] [分类讨论区] [全

I just got interview with Microsoft. One question they asked me was:

Suppose there is an interger array of size N, the value of each
element ranges from 1 to N-1, so there are at least two elements have
the same value. Find the fastest algorithm to detect the first
repeated element.

Enjoy!

```
class A
{
	virtual A();
	~A();
};

A::A() { printf("A"); }
A::~A(){ printf("~A"); }

class B : public A
{
	virtual B();
	~B();
}

B::B() {printf("B"); }
B::~B(){ printf("~B"); }

main()
{
	A * p = (A *)(new B);
	delete p;
}
```

what is the output?

There are n couples attending a party. Each one shakes hands with the persons
he doesn't know. (Assuming each person knows his/her partner) Mary and John
are a couple. John asked the rest of the party-attenders how many times he
has shaken hands. Each one gives a unique answer. How many times does
Mary shake hands?

I heard this from a friend.

You want to send a diamond to your friend in another city. All you have is a box which can be used with any number of locks, and you have all the locks available to you on earth. You know the postman between you and your friend is evil. He will try his best to get the diamond, but he doesn't want to leave any trace, meaning he will not break the box. The problem is that you have to send keys to your lock through this postman too.

Design a strategy to send the diamond to your friend safely.
This is a really boring question.

The recruiter asked me: How do you write malloc and free to detect memory reference violation?

Then she asked, supposed you want to flag a block of memory as used by putting some bit pattern at the beginning of the block. What bit pattern will you use?

There are some weights for a balance. Pick out four weights that can weigh largest range. By range I mean continuous weights starting from 1.

For example, 1, 2, 4, 8 are four weights that can weigh 1 through 15
This is from a friend working in Microsoft. He said he will use this question if he is going to interview someone.

Q: divide an interval into three equal parts using ruler and Yuan Gui
Suppose you have a single link list, but you don't have the head pointer.

Given a random pointer to a node in the list, delete this node from the list.

-- This is a question I told my manager to ask future interviewees.

There are two math genius. God chose a pair of consecutive positive integers and tells each genius one of the number. They don't know whether they get the larger number or the smaller number.

The only question they can ask each other is:
" Have you guessed my number?"

The only answers allowed is Yes or No.

They continued asking questions to each other. After a while, One of them answered yes. Explain why.

I gave this question to an Indian friend. He answered immediately. //faint
This question was proposed during one of our group meetings. It got all of
us into deep thinking...

John and Mary have 25 apples, Mary has 11 more apples than John. How many apples does Mary have? Why does she have more apples than John:-)
Here comes an easy one.

What is the next number in the sequence:

1, 2, 720!, ...
You might have heard of this:

Suppose there are two identical cups of water. One is red, the other is green. You take out 1ml of red water and put it into the green water. Then you take 1ml of (mixed) green water into the red water. Does the red water have more green or does the green have more red?

--
You might have heard of this:

You are given 20 blue balls and 20 red balls. You want to put them into two bags. Suppose someone will pick up a ball from one bag, we want you maximize the probability that the ball he picks up is red.