

Knapsack decision problem

```
bool knapsack(int weights[], int numweights, int total)
{ if (total==0)
  return true;
  if (numweights==0)
  return false;
  return knapsack(weights, numweights-1, total)
    || knapsack(weights, numweights-1, total-weights[numweights-1]); }
```

Number of ways of making change

```
bool changeways(int coinvals[], int numcoins, int value)
{ if (value==0)
  return true;
  if (numcoins==NULL)
  return false;
  return changeways(coinvals, numcoins-1, value)
    + changeways(coinvals, numcoins, value-coinvals[numcoins-1]); }
```

Satisfying a boolean function

(inputs to the function are kept in a bool array,
third parameter is number of inputs still undecided)

```
bool satisfiable(bool f(bool []), bool values[], int numunknown)
{ if (numunknown==0)
  return f(values);
  values[numunknown-1]=false;
  bool a = satisfiable(f, values, numunknown-1);
  values[numunknown-1]=true;
  bool b = satisfiable(f, values, numunknown-1);
  return a || b; }
```

Number of combinations

```
int combs(int n, int r)
{ if (r==0)
  return 1;
  if (n==0)
  return 0;
  return combs(n-1, r-1)
    + combs(n-1, r); }
```

Example of satisfying a bool function

```
bool testme(bool input[])
{ return ((input[0] && (!input[1] || input[2]) ||
          !(input[2] || input[3] && !(input[4] || input[1])))); }

void main()
{ bool values[5];
  if (satisfiable(testme, values, 5))
    cout << "The formula is satisfiable\n";
  else
    cout << "The formula is not satisfiable\n"; }
```

Is this integer a perfect square? (numbers represented as strings in binary)

```
bool isthishesquarerootofthis(string root, string num)
{ if (multiply(root, root)==num)
  return true;
  else
  return false; }

bool hasthisnumbergotasquarerootbeginningwith(string num, string start)
{ if (isthishesquarerootofthis(start, num))
  return true;
  return hasthisnumbergotasquarerootbeginningwith(num, start+"0")
  || hasthisnumbergotasquarerootbeginningwith(num, start+"1"); }

bool isthisaperfectsquare(string num)
{ return hasthisnumbergotasquarerootbeginningwith(num, ""); }
```