Extracting a solution and Hamiltonian Cycles

Lecture 23
CS 312

Objectives

- Extract the composition of a solution
- Apply the general backtracking scheme
- Solve the Hamiltonian cycle problem

Backpack (p.308)

```python
function backpack (items, remaining)
    bestSoFar := 0
    for choice = items+1 to numItems
        if weight[choice] <= remaining then
            bestSoFar := max (bestSoFar, value[choice] +
            backpack (choice, remaining - weight[choice]))
        return bestSoFar
```

Backtrack in general

```python
backtrack (V[1...k])
(V is a promising solution)
if V is a complete solution, then write V
else
    for each W[1...k+1] such that W[1...k] = V[1...k]
    and W is a promising solution
        backtrack (W[1...k+1])
```

Hamiltonian Cycles

- Given a graph with N vertices.
- Find a cycle that visits all N vertices exactly once and ends where it started.
- Sound familiar?
Example

Hamiltonian

```plaintext
Hamiltonian (k)
  while (1)
    x[k] = NextValue(k)
    if (x[k] < 0) then return
    if (k == N) then print solution
    else Hamiltonian (k+1)
  endWhile

NextValue (k)
  while (1)
    value = (x[k]+1) mod (N+1)
    if (value == 0) then return value
    for j = 1 to k-1 if G[j][value] = value then break
    if (j=k) and (k < N or k == N and G[N][x[1]])
    then return value
  endWhile
```

Homework

- Study for midterm
- Finish project 3 (due Wednesday the 14th)
- Read ahead?