Programming a Reduction

```
# This function should use the k_clique_decision function
# to solve the independent set decision problem
def independent_set_decision(H, s):
    return k_clique_decision(reform_graph(H), s)

if __name__ == '__main__':
    flights = [(1,2),(1,3),(2,3),(2,6),(2,4),(2,5),(3,6),(4,5)]
    G = {}
    for (x,y) in flights:
        make_link(G,x,y)

    for node in G:
        print node, "--",
        for x in G[node]:
            print x,
        print

    print
    print "independent set:"
    for i in range(1, len(flights)):
        print i, "-" , independent_set_decision(G, i)
```

Reduction: k-Clique to Decision

```
def k_clique(G, k):
    if not k_clique_decision(G, k):
        return False
    if len(G) is k:
        return G.keys()

    for edge in get_all_edges(G):
        break_link(G, edge[0], edge[1])
        if k_clique_decision(G, k):
            return k_clique(G, k)
        make_link(G, edge[0], edge[1])

if __name__ == '__main__':
    edges = [(1,2),(1,3),(1,4),(2,4)]
    G = {}
    for (x,y) in edges:
        make_link(G,x,y)

    for x in G:
        print x, G[x].keys()
    print

    k = 3
    print "k =", k
    print k_clique(G, k)
```

Polynomial vs. Exponential

Polynomial vs. Exponential

Theoreticians would say that an algorithm with a running time of n^{100} is efficient, but one with a running time of 1.1^n is not. Assuming these running times are exact, what's the smallest n for which the efficient algorithm is faster?

9623

From Clauses to Colors

From Clauses to Colors

In the reduction from 3-SAT to 3-COLORABILITY, we talked about a way of converting a 3-SAT problem with x variables and y clauses into a graph with n nodes and m edges. Give a formula for n and m. (Fill in the boxes to complete the equation. See the example given below.)

$$n = \boxed{0}\,x + \boxed{0}\,y + \boxed{0}$$
$$m = \boxed{0}\,x + \boxed{0}\,y + \boxed{0}$$

(ex. $n = 4x + 10y + 8$)

NP or Not NP?

NP or Not NP? That is the Question

Select all the problems below that are in NP. Hint: Think about whether or not each one has a short accepting certificate.

- [ ] **Connectivity**: Is there a path from x to y in G?
- [ ] **Short path**: Is there a path from x to y in G that is no more than k steps long?
- [x] **Fewest colors**: Is k the absolute minimum number of colors with which G can be colored?
- [x] **Near Clique**: Is there a group of k nodes in G that has at least s pairs that are connected?
- [x] **Partitioning**: Can we group the nodes of G into two groups of size $n/2$ so that there are no more than k edges between the two groups.
- [x] **Exact coloring count**: Are there exactly s ways to color graph G with k colors?