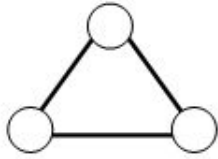
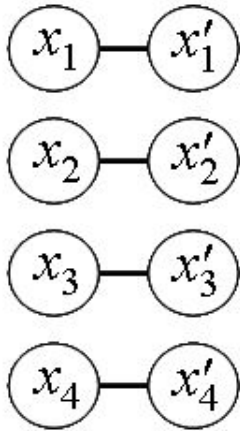


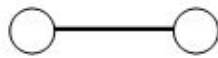


THEOREM OF THE DAY

Karp's Theorem (Detail) *Vertex Cover is NP-Complete.*



× 4




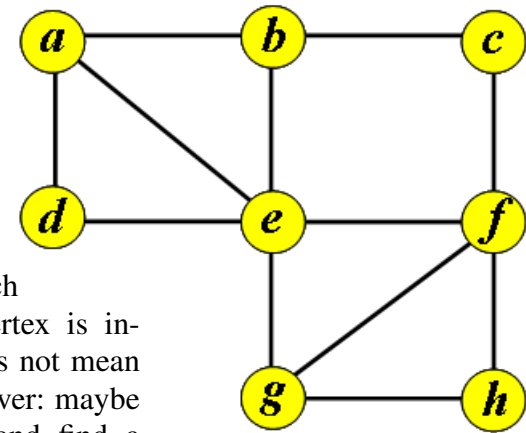
× 12



Image courtesy of **IKEA**

Given a graph G , for example the one on the right, can you find a *vertex cover*: a subset of vertices such that every edge is incident with at least one of these vertices? For this graph the subset $\{b, c, d, e, g, h\}$ is a cover which

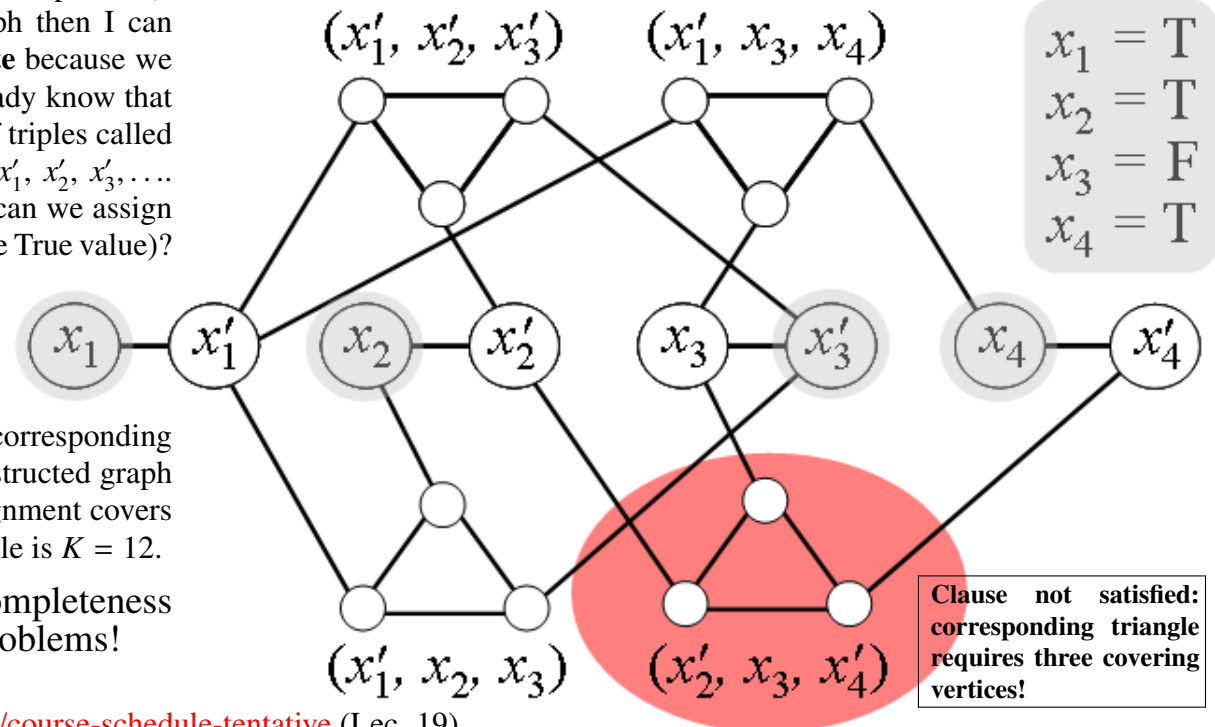
is *minimal*: every vertex is indispensable. This does not mean that it is a *minimum* cover: maybe you can start again and find a smaller? Certainly, the presence of two disjoint triangles  means that you cannot do better than 4 vertices because every triangle requires 2 vertices to cover it. So here is the problem known as **Vertex Cover**: given a graph G , and a positive integer K , can you find a cover of size at most K ? In our case the target might be set at $K = 4$. This is a **decision problem**: the answer is Yes or No.



A decision problem is said to belong to the class **NP**, roughly speaking, if evidence for a Yes solution can be checked easily (in a number of steps that is a polynomial function of the input size). Thus, if you assert that $\{a, c, e, g\}$ is a cover of size 4 for our graph then I can quickly spot that edge fh is not covered. Vertex Cover is **NP-complete** because we can transform 3-SAT problems to Vertex Cover problems, and we already know that 3-SAT is **NP-complete** (Cook, 1971). In 3-SAT we have a collection of triples called *clauses* containing logic variables, x_1, x_2, x_3, \dots , and their negations x'_1, x'_2, x'_3, \dots . If x_i is True then x'_i is False and vice-versa. The decision problem is: can we assign truth values to each x_i so that each clause is *satisfied* (contains at least one True value)?

For example, $(x'_1, x'_2, x'_3), (x'_1, x_3, x_4), (x'_1, x_2, x_3), (x'_2, x_3, x'_4)$, is satisfied by $x_1 = F, x_3 = T$, with arbitrary values for x_2 and x_4 . How is this instance of 3-SAT transformed into Vertex Cover? We take a triangle for each clause and a single edge for each pair x_i, x'_i , as shown above. A further 12 linking edges join clause entries to their corresponding single edge vertices, as shown on the right. The resulting cleverly constructed graph has a cover with just 2 vertices per triangle if and only if a 3-SAT assignment covers the third linking edge to each triangle. The target K value in our example is $K = 12$.

A classic 1972 theorem of Richard Karp asserts the **NP-completeness** of Vertex Cover and no fewer than twenty other decision problems!



Clause not satisfied: corresponding triangle requires three covering vertices!

Web link: sites.google.com/site/cs172spring2011/course-schedule-tentative (Lec. 19).

Further reading: *50 Years of Integer Programming 1958-2008* Michael Jünger et al (eds.), Springer 2010. Karp's 1972 paper is reproduced, with a new preface by him, as Chapter 8.

