

Hints for Power Gems

- We have n gems. We also have m *possible* links between pairs of gems, each with a power value. How can we represent this as a graph?
- The statement describes a piece of jewellery as a set of gems, linked directly/indirectly. What does this correspond to, in our graph model?
 - We can choose which gems to take.
 - Amongst the possible links between the taken gems, we can choose which ones to actually use, as long as all chosen gems are linked directly/indirectly.
- There is a tradeoff between making bigger jewellery (needing more links, possibly of worse power), and higher-quality jewellery (uses less-available high-power links).
 - Maximum quality is a non-increasing function of required minimum size.
 - Maximum size is a non-increasing function of required minimum power.
- We can examine the dual problem: Instead of best-quality for at-least-size- g_i , we can flip it around and look at largest-size for at-least-quality- q .
 - If we want quality q , which possible links can we use, and which are forbidden?
 - Which of those links should I choose?
 - * The gems must be linked together in the end, so we need to add enough links to join every single gem onto the piece of jewellery.
 - * Does it help us to add **extra** links, between already linked gems?
- If we can solve the dual problem, we can solve the original problem for some minimum size g_i .
- We can slowly decrease the quality threshold, which results in larger maximum sizes.
 - Lowering the quality threshold allows us to use some new links. Which of these should we actually use?
 - If we see the max size exceeds g_i for the first time, what does that say about the corresponding quality threshold?
 - The range of quality threshold values is quite wide, so going +1 at a time will be slow. Which particular threshold values actually matter?
- What algorithm have we seen before, does the exact same thing?
- We want to answer many queries with different g_i values. Can we compute all possible answers, in a single processing step?