**Background**

The Android Spotify client uses a lot of images that come from our backend servers. That said, a layer of caching for images is used to avoid unnecessary trips to the back end servers. You have been tasked with implementing and designing an in-memory least-recently-used (LRU) image cache.

**API**

The cache should support setting a limit on the number of images it holds. Choose a suitable default.

The cache should support setting a limit on the total number of bytes taken up by the images it holds. Choose a suitable default.

The cache should evict images from the cache if upon adding an image either of the above two limits is exceeded.

The keys for the images are 32-byte buffers. Choose a suitable representation.

**Requirements**

Comments should be written in English.

Document your assumptions.

Your code should compile and It should have appropriate unit tests.

**Documentation**

The code and design decisions should be sufficiently documented that an experienced developer could use it. Choose a suitable format.

**Go the extra mile**

We've asked you to use the least-recently-used eviction policy. But one could conceivably evict from a cache on other criteria such as image size. For extra credit, make the eviction policy generic so that an experienced developer could write his or her own eviction policy and still use your image cache implementation.