در این پروژه به کمک شی گرایی کلاسی تحت عنوان conversionInfo ساخته ایم که آبجکتهای آن اطلاعات مربوط به تبدیل یک واحد پول به واحد پول دیگر را نگهداری میکنند. که این اطلاعات شامل شماره index واحد پول مبدا و مقصد و ضریب تبدیل آنها به یکدیگر است. همچنین مسیری که پرسود ترین تبدیل را شامل میشود نگهداری میکنیم. کار دیگری که انجام دادیم این است که برای راحتی محاسبات از ضرایب تبدیل لگاریتم گرفته ایم و حال بجای اینکه هر بار ضرایب را در هم ضرب کنیم فقط کافی است لگاریتم آنها را با هم جمع کنیم. همچنین باید به این نکته دقت کنیم که اگر ضرب دو عدد 1 بشود، جمع لگاریتم آنها برابر 0 میشود.

در این پروژه همچنین باید از یک الگوریتم برای یافتن کوتاهترین مسیر بین یالهای یک گراف جهتدار استفاده میکردیم. الگوریتم های دایجکسترا و بلمان فورد برای یافتن کوتاهترین مسیر از یک راس به راسهای دیگر کاربرد داشت ولی ما میخواستیم کوتاهترین مسیر موجود بین تمامی راسها را پیدا کنیم، با این وجود اگر گرافمان تنک بود و یالهای زیادی نداشت بهتر بود الگوریتم دایجکسترا را به تمامی راس ها بسط میدادیم و برای همه آنها اجرا میکردیم. اما در این مورد مثال چون تعداد یالها از O(2|v|)  
است مناسب بود که از الگوریتم Floyd-warshall استفاده میکردیم که کوتاهترین مسیر بین تمامی دو جفت راسها را پیدا میکند.

همچنین پیچیدگی زمانی کل مسئله هم وابسته به الگوریتم فلوید وارشال است. که پیچیدگی زمانی این الگوریتم هم از O(|V|^3) است. پس پیچیدگی زمانی کل مسئله هم از O(|V|^3) است.

توضیح مختصری از الگوریتم فلوید وارشال: این الگوریتم در واقع درون 3 تا حلقه انجام میشود که که با هر بار تکرار حلقه یکی از اعداد i,j,k زیاد میشوند. این الگوریتم میگردد که اگر مسیری کوتاهتری بین i و j وجود داشت آن را پیدا کند و در واقع آن k هم شماره راسی است که برای یافتن مسیر کوتاهتر از آن گذر میکنیم.

توضیحات: برای حل این مسئله در اینترنت جستجو کردیم و از ایده های موجود استفاده کردیم که با توجه به آنچه خواندیم به نظر میامد این راه بهترین راه موجود برای حل مسئله arbitrage و یافتن پرسودترین راه (بدون تکرار و حلقه) است.